Reprinted with permission from American Laboratory, Vol. 7, No. 12, pp. 75-77.

Microcomputer output instruction

By David G. Larsen, Peter R. Rony, and Jonathan A. Titus

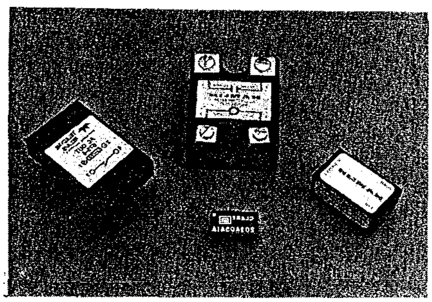

Figure 1a *Typical optically isolated solid-state relays (clockwise from left); 1.5-A, 120-v ac relay, model P/N 701-11-5 [Hemlin}: 10-A, 220-v ac relay, model* 7522 *(Hamlin); 3-A, 120-v ac relay, model* 675-4 *(Teledyne); and 0.75-A, 120-v ac relay, model 203A05AIA (Clare).*

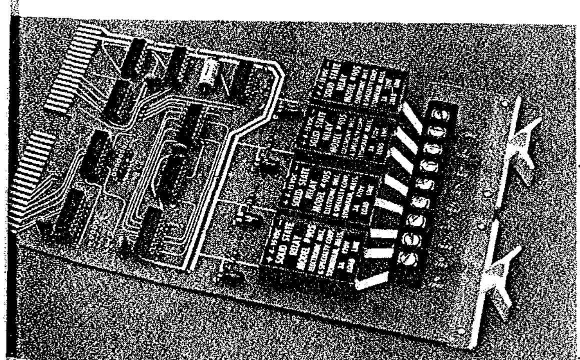

Figure 1b *A printed circuit board containing four 3-A, 120-v ac relays (Electronic Instruments and Specialty Corp.} and all necessary external device addressing circuitry.*

IN A PREVIOUS COLUMN, we discussed different types of simple input/output devices and provided a listing of general principles of interfacing that apply to a wide variety of computers. This month, we would like to explain how computer instructions, i.e., software, cause an I/O device to operate.

Controlling power with a microcomputer

The I/O device that we shall choose for our discussion is the optically isolated solid-state ac relay. These relays can control any ac power device within the output current ratings of the relay. Typical solid-state relays are shown in *Figures 1a-b.* Such relays permit a single TTL output signal of logic 0 or logic 1 to control up to 10A of 220-v ac power, as is possible with the Hamlin model 7522 relay shown at the top middle part of Figure 1a. Internally, each relay contains a light-emitting diode, a light-sensitive transistor, a power triac, and a transparent dielectric optical path that isolates the digital and power circuitry and can itself withstand a voltage difference of at least 1000 v.

A typical microcomputer I/O circuit that employs the solid-state relay is shown in *Figure* 2. You will recall from a previous column that the microcomputer sends synchronization pulses, called *device select pulses,* to the I/O device. In the figure, these are the ‾|_|‾ pulses from the SN74154 decoder circuit that, for an 8080 microcomputer, each have a time duration of only 500 nsec. It should be clear that a single 500-nsec pulse cannot sustain the continuous operation of an ac power device. What is required is a simple interface between the microcomputer and the solid-state relay that would permit the ac power device to operate continuously, if it is so desired. A suitable interface is a single SN7474 positive-edge triggered flip-flop and a single buffer from a hex buffer/driver chip. The buffer is needed since it is not good engineering practice to drive a solid-state ac relay directly from the output of a flip-flop.

With the aid of a suitable program, the microcomputer and SN74154 decoder can generate individual device select pulses that either clear or set the SN7474 flip-flop. *To clear the flip-flop, and thus turn on the ac power device, only a single 500-nsec pulse is needed.* The flip-flop output, *Q,* will remain at logic 0 until a single 500-nsec pulse is applied to the preset input, at which time the ac device will turn off.

It should be noted that any simple open collector gate or inverter can be used as the buffer between the output of the flip-flop and the input of the solid-state relay. Suitable choices would be the SN7401 or SN7403 2-input NAND

Mr. Larsen, Department of Chemistry, and Dr. Rony, Department of Chemical Engineering, are with the Virginia Polytechnic Institute and State University. Mr. Titus is Consultant, Tychon, Inc.

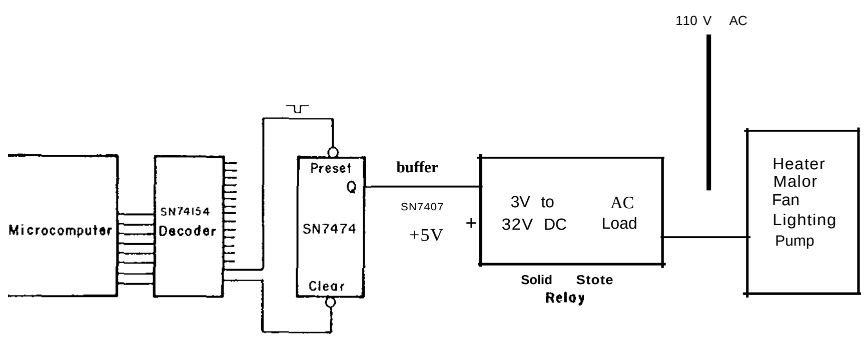

Figure 2 *A typical I/O circuit for a power ac device such as a fan, heater, motor, or antenna control.*

Table 1

Program incorporating out instruction

Memory address	Instruction byte	Description
O	11010011	Send device select pulse to device given by the following 8-bit device code
	00000000	Device code for clear input to SN7474 flip-flop
2	01110110	Halt the microcomputer

Table 2

Program incorporating off instruction

Memory address	Instruction byte	Description
O	11010011	Send device select pulse to device given by the following 8-bit device code
	00000001	Device code for preset input to SN7474 flip-flop
2	01110110	Halt the microcomputer

Table 3

Program incorporating additional instructions

Memory address'	Instruction byte	Description
23	11010011	Turn solid-state relay on, i.e.; send device select pulse to the device given by the following 8-bit device code
24	00000000	Device code for clear input to SN7474 flip-flop. When the flip-flop is cleared, the solid-state relay turns on.

At this section of the program various points might arise that determine whether or not the solid-state relay is turned off. Typical questions to be resolved include: 1) Has sufficient time elapsed? 2) Has the reaction proceeded to completion? 3) Has the piece of machined metal reached its final desired dimension? 4) Is the temperature too high? or 5) Have all of the peaks been eluted from the chromatograph?

Memory address'	Instruction byte	Description
107	11010011	Turn solid-state relay off, i.e., send device select pulse to the device given by the following 8-bit device code
120	00000001	Device code for preset input to SN7474 flip-flop. When the flip-flop is set, the solid-state relay turns off.

(Additional program steps)

gates, the SN7405 inverter, or the SN7409 2-input AND gate.

Output instruction

Seventy-eight different instructions and a total of 256 variations of such instructions exist for the 8080 microprocessor chip. Each instruction contains a single 8-bit *instruction code* that indicates which type of operation or group of operations the microcomputer will execute. Some instructions contain two or three 8-bit bytes that are present in successive memory locations. A *byte* is defined as a group of eight contiguous bits occupying a single memory location.! Thus, 8080 microprocessor instructions are either 8, 16, or 24 bits long, with the first eight bits always being the instruction code.

The OUT instruction is a 16-bit instruction that consists of two successive 8-bit bytes located in successive memory locations. The first byte, in binary code, is always 11010011_2. The second byte can be any 8-bit binary number from 00000000_2 to 11111111_2; this is the device code of the specific output device that will receive eight bits of data from the accumulator. The instruction can be summarized as follows:

11010011_2 $XXXXXXXX_2$

Generate a device select pulse, with the aid of an 8-line to 256-line decoder circuit, to allow an 8-bit data byte present in the accumulator to be sent to the desired output device. The contents of the accumulator remains unchanged.

Simple programs

The simplest program that incorporates the 0 UT instruction is probably the one shown in *Table 1*. An 8080 microcomputer operating at a clock rate of 2 MHz will execute this program in 8.5 *usee.* The ac power device will remain on once the program has been executed.

To turn off the device, a slightly different program is required, as shown in *Table* 2. The ac power device will turn off after the second instruction byte in the program and remain off after the microcomputer halts. A more practical program requires additional instructions. Several such programs can be found in Ref. 1. Many of them have the basic form shown in *Table 3*.

Keep in mind that a memory address contains 16 bits. When we write "memory address 0," we really mean the memory address corresponding to the following 1e-bit binary word: 00000000 00000000. Note that the sixteen bits have been split into two parts, the most significant eight bits and the least significant eight bits. These are called the HI (or H) and LO (or L) memory addresses, respectively.

With the aid of the program in Table 3, the solid-state relay shown in Figure 2 will turn on and off according to various decisions made by the program. A typical microcomputer-controlled system could easily have several such relays.

In a more orderly and systematic treatment of the 8080 microprocessor, one would probably introduce the 8080 instruction set prior to the discussion of any particular instruction, such as the 0 UT instruction described this month. Since we consider it more beneficial to the reader to deal with the 0 UT instruction, we have decided to treat it first. Microcomputer instructions will be discussed in considerable detail in subsequent columns. In the next column, we will explain how the 8-line to 256-line decoder circuit shown in Figure 2 generates individual device select pulses.

Reference

1. *Bugbook Ill. Microcomputer Interfacing Experiments Using the Mark 80® Microcomputer, an 8080 System* (E & L Instruments, Inc., Derby, Conn., 1975).