Behavior Research Methods, Instruments, & Computers
1993, 25 (4), 477-478

An interface for controlling external devices via the IBM PC/XT/AT parallel port

MICHAEL R. MARKHAM
University of New Mexico, Albuquerque, New Mexico

This article describes the construction and operation of a simple interface to control external devices via the IBM PC/XT/AT's parallel printer port. The interface is relatively inexpensive and easy to build. It is potentially useful for controlling behavioral experiments.

Several reports in this journal have discussed sources of input to the IBM PC. These included using the keyboard, mouse, game port, and parallel port as input devices (Crosbie, 1990; Dalrymple-Alford, 1992; Segalowitz & Graves, 1990). In addition to accepting input, it is often desirable to control external devices (e.g., slide projectors, pellet dispensers) with the IBM PC when conducting behavioral research. This article is a description of an interface that can be used to control up to eight external devices via the IBM PC's parallel printer port. The interface is inexpensive (the retail cost of parts for the interface is less than $45), and takes about 2 h to build.

For each parallel port, Pins 2-9 of the DB-25 female connector are the data output pins for that port (Eggebrecht, 1990). Data is sent through the port in binary format, 1 byte at a time. Thus, when values are output through the parallel port, Pins 2-9 are set high (5 V) or low (0 V). These outputs can be used to turn relays on or off in order to control devices that are operated by switch closures. Because the output current of the parallel port is insufficient to power most relays, controlling relays directly with the parallel port would risk damage to the circuitry of the port. However, the output of the parallel port can be used to control a 74LS125 solid-state switching buffer. This switching buffer can then be used to turn relays on and off according to the output of the parallel port. An external power source supplies the power to turn the relays on and off.

Construction of the Interface

The schematic diagram for the interface is shown in Figure 1. The circuit can be constructed by any convenient means (e.g., solderless breadboard, wire wrapping, etc.). The relays (R1-R8) can be any relay with a 5-V dc coil and a *minimum* coil resistance of 200Ω. The maximum current rating of each relay's contacts should be sufficient for the current that will be switched by that relay. For example, if a relay is used to control a pellet dispenser with a current rating of 1.75 A, the contacts of the relay should be rated at 2.0 A or more. The diodes (D1-D8) are essential in order to prevent damage to the 74LS125 by inductive kickback from the relay coils. Any convenient dc power supply (e.g., an ac wall outlet adaptor) with a voltage of +7.5 to +9 V and a minimum current of 500 mA can be used to power the interface. The 7805 regulates power to the circuit at +5 V. To dissipate heat generated by the voltage regulator, an appropriate ventilated heatsink should be attached to the 7805.

Software Control of the Interface

Relays 1-8 can be opened and closed by sending appropriate values to the parallel port. Each of the relays is controlled by 1 bit of the 8-bit parallel port output. When that bit is set to 0, the relay is turned on (closed). When the bit is set to 1, the relay is turned off (open). Thus, a value of FF (hex) or 255 (decimal) sent to the parallel port would turn all relays off, whereas a value of 0 would turn all relays on. Output is sent to parallel port LPT1 by assigning values to the I/O address 0378 (hex), or to parallel port LPT2 by assigning values to the address 0278 (hex).

When the interface or the computer is first turned on, any or all of the relays might be on. Therefore, it is necessary to turn off all the relays before turning on any devices controlled by the interface. Sending a value of FF (hex) or 255 (decimal) to the parallel port turns all the relays off. If the interface is connected to parallel port LPT1, the Turbo Pascal command for this is:

```
Port[$0378] := $FF;
```

Once the interface is initialized in this manner, the relays can be turned on and off by sending appropriate values to the parallel port. To turn a relay on, subtract its bit value from FF (hex) or 255 (decimal) and send the result to the parallel port. The bit values for all relays are shown in Table 1. Whenever a new value is output to the parallel

Correspondence should be sent to M. R. Markham, Department of Psychology, Logan Hall, University of New Mexico, Albuquerque, NM 87131-1161 (e-mail: mmarkh@unmb). I would like to thank Pat Sharp for his invaluable assistance in the design of the interface described in this paper. Thanks also to William B. Cushman, E. C. Dalrymple-Alford, Roger Graves, and one anonymous reviewer for their helpful comments on the design of this interface and an earlier draft of this manuscript. I was supported by a Ford Foundation Predoctoral Fellowship during the preparation of this manuscript. Development of the interface described herein was supported in part by a grant from the Research Allocations Committee at the University of New Mexico to Michael J. Dougher.

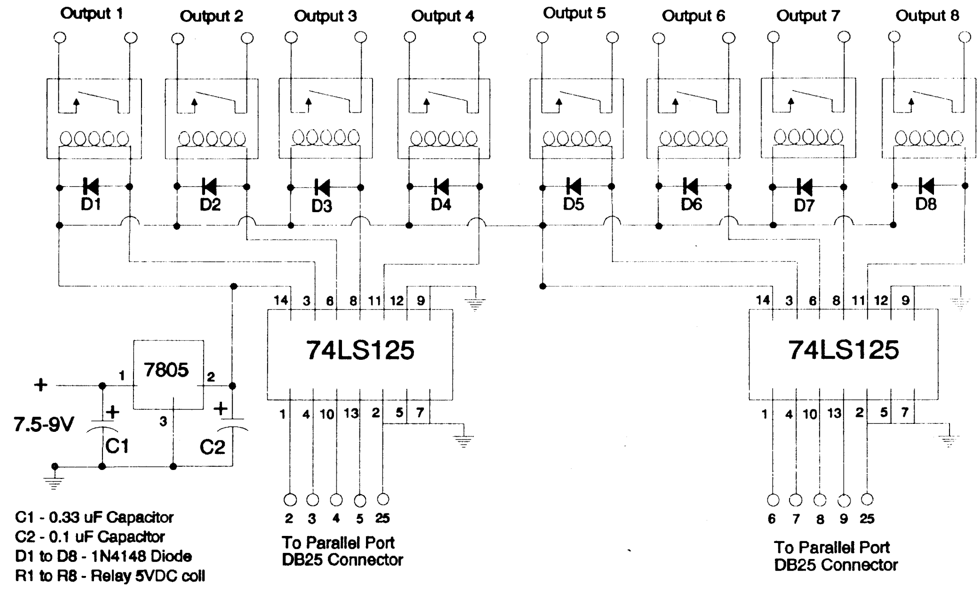

Figure 1. Schematic diagram for the interface.

port, all relays not specifically turned on will be turned off. Once a relay is turned on, it will remain on until turned off. For example, the following Turbo Pascal code will turn on Relay 6, then turn it off after 500 msec:

```
Port[$0378] := ($FF - $20); { Turn Relay 6 ON }
Delay(500);                 { Wait ½ second }
Port[$0378] := ($255);  { Turn all switches OFF }
```

In the above example, Relay 6 will be turned on, and all other relays will be turned off. Then, after 500 msec, all relays will be turned off.

To turn on multiple relays simultaneously, subtract the sum of their bit values from FF (hex) or 255 (decimal) and send the result to the parallel port. For example, to turn on Relays 1, 5, and 8, sum the bit values of the relays: $(1+16+128) = 145$, then subtract the result from 255: $255-145 = 110$. Sending the resulting value of 110 (decimal) or 6E (hex) to the parallel port would turn on Relays 1, 5, and 8.

Additional Design Considerations

This interface provides an inexpensive and simple way to control external devices via the IBM XT/AT parallel port. In addition to controlling external devices, the parallel port can also be used for response key input, as suggested by Dalrymple-Alford (1992). Thus, by accepting input and controlling output via the parallel port, the IBM PC/XT/AT parallel port can serve as a simple and inexpensive means of controlling behavioral experiments.

There are other integrated circuits that might be used to design interfaces similar to the one described here. For example, the 7406 and 4066 devices would also be suitable for such an interface, and they offer certain advantages such as the capacity to control larger relays. When using an interface such as this, it is important to remember that relays have a nonnegligible switching time—sometimes several milliseconds—that must be considered when precise timing is important. Using high-speed relays, such as reed relays, will minimize this switching time. In cases in which very high-speed switching is required, it is possible to control low-current devices (less than 24 mA) operating at 5 V by connecting them directly to the 74LS125 buffer.

Table 1
Bit Values for Controlling the Relays

Bit Value		
Decimal	Hex	Controls Relay
1	01	1
2	02	2
4	04	3
8	08	4
16	10	5
32	20	6
64	40	7
128	80	8

REFERENCES

Crosbie, J. (1990). The Microsoft mouse as a multipurpose response device for the IBM PC/XT/AT. *Behavior Research Methods, Instruments, & Computers*, **22**, 305-316.

Dalrymple-Alford, E. C. (1992). Response-key input via the IBM PC/XT/AT's parallel printer port. *Behavior Research Methods, Instruments, & Computers*, **24**, 78-79.

Eggebrecht, L. C. (1990). *Interfacing to the IBM personal computer* (2nd. ed.). Carmel, IN: H. W. Sams.

Segalowitz, S. J., & Graves, R. E. (1990). Suitability of the IBM XT, AT, and PS/2 keyboard, mouse, and game port as response devices in reaction time paradigms. *Behavior Research Methods, Instruments, & Computers*, **22**, 283-289.

(Manuscript received February 8, 1993;
revision accepted for publication July 26, 1993.)