COMPUTER TECHNOLOGY

Laboratory functions for the Tecmar Lab Tender and IBM PC

GARY FINLEY

The University of Alberta, Edmonton, Alberta, Canada

A set of assembly language functions has been written to interface laboratory software for the IBM PC to the Tecmar, Inc., Lab Tender circuit board. These functions provide convenient access to the Lab Tender's parallel ports, timers, A/D converters, and D/A converters from laboratory control programs written in the C language.

This paper describes a series of assembly language functions that provides a software interface between programs for the IBM PC microcomputer and the Lab Tender circuit board made by Tecmar, Inc. (23600 Mercantile Road, Cleveland, Ohio 44122). The Tecmar Lab Tender occupies one slot of the IBM computer's expansion bus and contains laboratory interface hardware, including three 8-bit parallel data ports, five 16-bit timers, and several channels of both digital to analog (D/A) and analog to digital (A/D) data conversion. These components allow a Lab Tender-equipped IBM PC to control both digital laboratory equipment, such as operant-chamber food dispensers and optical shutters, and analog components, such as dc light sources and variable-speed motors. The Lab Tender allows the IBM PC to measure both analog laboratory signals (e.g., the electrical waveforms of biofeedback research) and digital data (e.g., the condition of response switches used in reaction-time experiments).

This laboratory software is written for the IBM PC in the C language instead of the interpreted BASIC language supplied with the computer. Compiled C programs run much faster than interpreted BASIC programs do, and it is easier to produce well-organized programs in C than in BASIC. We use the C software development system sold by C Ware Corp. (P.O. Box C, Sunnyvale, California 94087). As a convenient software interface between the C programs and the resources of the Tecmar Lab Tender, we have written a set of functions in 8088 assembly language that perform some standard operations with its parallel ports, timers, and analog input/output (I/O) hardware. This set of functions, called TECIO.A, may be of interest to others who use the IBM PC and Tecmar Lab Tender in laboratory research, even if their choice of language compilers is different from ours. With some modifications, TECIO.A should be usable with any FORTRAN, Pascal, or C software development system, provided an assembler is available, and code modules

produced by the language compiler can be linked to assembly language modules.

FUNCTIONS INCLUDED IN TECIO.A

TECIO.A contains three functions for handling the Lab Tender's parallel ports, five functions for timing tasks, and one function each for the A/D and D/A converters. Included in the listing's comments are descriptions of each function's actions and examples of the correct syntax for calling each function from a C program. The comments also include a brief discussion of the method of passing arguments between C programs and assembly language routines.

The Parallel Port Functions

Parallel I/O on the Lab Tender is provided by an Intel 8255 which contains three general purpose 8-bit ports. These ports can be configured to operate in a number of modes which use the port bits in various combinations for data input, data output, and control signal functions. The individual operating modes are enabled by storing appropriate data into the 8255's MODE register, as described in the Lab Tender manual. The parallel ports can be put into a given operating mode by calling the function TECSET(M) with the appropriate MODE register data as the argument M. Once the operating mode has been set, data can be read from the parallel ports with the function TECI(P), where P specifies the address of the port (A, B, or C) to be read. Data is written to the ports with the function TECO(P,D), where P is the port address as above, and D is the byte of data to be output.

The Timing Functions

The Lab Tender performs timing and event counting tasks with an Advanced Micro Devices, Inc., Am9513 which contains five 16-bit programmable timers. The TECIO.A function DEL configures Am9513 timer 4 as a binary down-counter that is loaded with the value specified in the function argument. This count is decremented under the control of a 1000-Hz oscillator, and DEL ter-

The author's mailing address is: The University of Alberta Psychology Department, Edmonton, Alberta, Canada T6G-2E9.

minates when the count reaches zero. Thus DEL(100) will delay program execution for 100 msec. The 16-bit counter register can hold values up to 65535, so DEL can produce delays of up to 65.535 sec duration. Stopwatch-type timing functions are initialized by a call of the function INITCLK, which zeros the Am9513 timer 3 counter register and configures this timer as a binary up-counter incremented by a 1000 Hz oscillator. After INITCLK has been called, timer 3 can be started and stopped in stopwatch fashion by calls of the functions STRTCLK and STOPCLK. The accumulated count of milliseconds can be read from the timer 3 counter register at any time by calling the function READCLK.

The Analog Signal Functions

The Lab Tender contains a 32-channel 8-bit A/D converter for the digital measurement of external analog voltages. TECIO.A accesses this device through the function ATOD(N), where N is the number (from 0-31) of the A/D channel to be read.

The Lab Tender contains a single D/A converter connected to an array of 16 sample-and-hold amplifiers which act as a system of analog voltage latches. This combination mimics the behavior of 16 separate D/A converters, because the single converter can be used to store different voltages in each of the 16 separate latches. TECIO.A accesses the D/A converter system through the function DTOA(L,D) where L specifies the analog latch to be set, and D is the 8-bit digital value to be converted.

CONCLUSION

The functions in TECIO.A provide a few basic operations to expedite the use of the combination of the IBM PC and the Tecmar Lab Tender in laboratory computing applications. For situations which require special-purpose functions, the contents of TECIO.A can serve as examples of the technique. TECIO.A demonstrates one approach to the task of accessing IBM PC I/O bus devices from application programs, an approach that has the advantage of program modularity. If specialized device interface code like that assembled from TECIO.A is held in separate object code modules, a program that calls these routines does not have to contain the device interface software at the source level. This saves disk space, decreases compiling time, and makes the program more compact and legible.

AVAILABILITY

The author will provide a printed listing of TECIO.A and some example C programs which call its functions on request. The files will be copied to an IBM PC DOS-format 5.25 in. floppy disk if one is included with the request. Write to the author at The University of Alberta Psychology Department, Edmonton, Alberta, Canada T6G-2E9.

(Manuscript received February 12, 1985;
revision accepted for publication May 8, 1985.)