FIVE LABORATORY INTERFACING PACKAGES

BY PATRICIA WIRTH AND LINCOLN E. FORD

EDITOR'S NOTE: This article is not a review in our usual sense. To understand the intent of the review, you must understand how it came to be written. Dr. Ford is a medical researcher at the University of Chicago. He wrote an extensive driver for the Tecmar Lab Master data-acquisition board and wanted to publish a description of it. The program is a threaded interpretive language interfaced to BASIC and embodies an essentially complete set of commands for controlling the Lab Master board. As a condition to publishing his description of the language ("SALT," June 1985 BYTE, with Samuel D. Fenster, who wrote most of the code), we prevailed upon Dr. Ford to make the program and its documentation available to our readers at the nuisance-value cost of filling orders ($50).

There are advantages and disadvantages to using programs written by researchers instead of commercial products. The primary disadvantage is that the program is written to do what the researcher needs to do, not to support the requirements of a large group of customers. In the case of Dr. Ford's program, his requirements are far-reaching, so SALT ends up supporting most Lab Master functions. For example, while he had originally used only the 100-kHz option of the board, he later acquired a 30-kHz board and extended the program to include that configuration. However, he does not use other boards, so he does not support them. Similarly, Dr. Ford does little on-line analysis with his data, so the routines for such functions are limited.

The primary advantage of using a program developed by a researcher instead of by a programmer is that a programmer debugs programs, while a researcher debugs experiments. For example, Dr. Ford points out that all the commercial packages that he examined are unable to synchronize exactly the A/D board timers with external-event triggers. This kind of problem can be critical in an experiment but will not show up as a program bug.

With all this in mind, we asked Dr. Ford to take a look at some of the commercially available software packages for running A/D boards. The question we posed to Dr. Ford was, Given that SALT is available for $50, why would anyone buy the commercial packages? Or, less sanguinely, what are the special strengths of these packages that make each of them uniquely attractive? What follows is the assessment by Dr. Ford and his research associate Patricia Wirth of what each package is best suited to do.

In the past two years, several general laboratory interface devices for microcomputers have appeared on the market, as well as a selection of software packages for running them. This review is aimed at the software packages, specifically at those products designed for use with two separate interface boards for the IBM Personal Computer.

Our perspective is that of physiologists. We acquire large blocks of digitized data, often controlling various aspects of the experiments at the same time that the data is being collected. In general, we have given a lower priority to on-line data analysis and to graphics, since we do most of our analysis after the experiments. At the other extreme, one of the products reviewed here, ILS from Signal Technology Inc., was developed primarily as a data-analysis package, with relatively little emphasis on laboratory control. Thus, the diversity in the software packages to be reviewed will accommodate many laboratory requirements, although not all specialized needs will be fulfilled by the same package. In this review, we have emphasized the data-acquisition and laboratory-control aspects of the packages.

INTERFACE BOARDS

All the software packages will control one or both of two interface boards, the Tecmar Lab Master and the Data Translation DT2801. See table 1 for the characteristics of the two boards. The major differences are that the DT2801 has a provision for direct memory access that the Lab Master does not, while the Lab Master has programmable counters not available with the DT2801. The direct memory access allows faster data transfers. The programmable counters on the Lab Master operate independently of the central processor, so accurate timing and control of experiments can be carried out much more rapidly than with the DT2801.

Both boards have a series of options. In addition to the standard characteristics listed in table 1, we have a DT2801 with a 16-bit A/D converter. This provides much higher resolution at the cost of a decrease in the maximum rate per conversion from 13.7 to

(continued)

Patricia Wirth is a Ph.D. student at Northwestern University who is collaborating in biomedical research at the University of Chicago. Lincoln E. Ford, M.D., is associate professor of medicine and cardiology at the University of Chicago. Address all correspondence in care of Dr. Ford at Hospital Box 249, University of Chicago Hospitals, 950 East 59th St., Chicago, IL 60637.

2.5 kHz. We have also obtained a Lab Master board with a 100-kHz option. Some other options that we have not tried are listed in table 1.

SOFTWARE

A list of the packages reviewed and the major characteristics of each package appears in table 2. LABPAC and PCLAB are sold by the manufacturer of the Lab Master and DT2801, respectively, and so are intended for use with those boards only. In addition, ILS will operate only the DT2801. We wrote our own language, SALT, for only the Lab Master. The two remaining packages, ASYST (Macmillan Software Co.) and Labtech Notebook (Laboratory Technologies), have provisions for running both boards, as well as A/D boards from IBM, Cyborg, Keithley DAC, and MetraByte. In addition, Labtech Notebook will support Acrosystems, Burr-Brown, Datel, Action Instruments, Coulbourn Instruments, Interactive Microware, Advanced Peripheral, and Datatek Microsystems boards.

The commercial packages cost between $249 and $1490 for the laboratory-application portion of the program. Two of them have separate, additional modules for graphics and data analysis, each module costing about $500 to $1500, so it is possible to spend up to $2500 for a total package.

One major difference among the packages reviewed is the language they use. The two provided by the manufacturers of the boards simply use calls to assembly-language routines from higher-level languages. Instructions for experimental functions are thus written as part of a larger program in the higher-level language. This can be a great benefit when you require on-line computation during the experiments, since you can program these computations in a highly structured, well-documented, and well-known language. Another advantage of this approach is that the data is stored in files that the higher-level language can read later, when future analysis is required. The disadvantage with BASIC as the higher-level language is that the calls to assembly language routines are slow, requiring about 2 milliseconds, so substantial delays can occur when several functions are to be run in sequence. This disadvantage is not as great when you use a compiled language, such as FORTRAN or Pascal, but these languages have the disadvantage inherent in any compiled language, namely, that they are not interactive, and you lose control of the program once you start it. You can stop an interpretive language such as BASIC anywhere in its operation without losing control. Once you stop the program, you can determine or change the value of a variable and change the direction of program flow. This provision is a major advantage in the laboratory, where you cannot always anticipate events external to the computer in advance of the experiments.

Two packages, Labtech Notebook and ILS, operate on single commands at a time. Labtech Notebook is entirely menu-driven, while ILS simply accepts single commands. These packages are easy to get started. ILS creates files that FORTRAN can read, while Labtech Notebook creates files that Lotus 1-2-3 can read.

A third approach is the use of a separate, specialized language for operating the laboratory functions. A form of languages especially suited to this type of operation is the threaded interpretive languages (TILs). These languages consist of a dictionary of routines, "words," for carrying out specific tasks. Primary words, called primitives, consist of machine language subroutines. More complex words, secondaries, consist of a sequence of calls to the primitives. Higher-level words may call primitives or previously defined secondaries in

(continued)

Table 1: *Comparison of Data Translation* DT2801 *and Tecmar Lab Master A/D interface board functions.*

	DT2801	Lab Master
Number of A/D channels	16 standard	16 standard—expandable to 256 optional
Resolution	12-bit standard 14- and 16-bit optional	12-bit standard 14- and 16-bit optional
Speed	13.7 kHz standard 27.5 kHz optional 2.5 kHz with 16-bit resolution	30 kHz standard 40 and 100 kHz optional 10 kHz with 14-bit resolution
Gain	Software-selectable up to 8 times on standard board Up to 500 times optional with substantial slowing of A/D rates	Software-selectable up to 500 times optional Resistor-selectable Gains up to 2000 times are available
Number of D/A channels	2	2
Digital I/O	16 channels configured as input or output in two 8-channel ports	24 channels configured as input or output in three 8-channel ports or two 8-channel ports and two 4-channel ports
Clock	400-kHz clock divisible by powers of 2 up to 2^{16} (163.8 seconds)	1-MHz clock divisible by powers of 10 up to 10^4 (10 milliseconds) or by powers of 2 up to 2^{16} (65.536 seconds)
Programmable timer/counters	None	5 separate counters with outputs that can be made to change without intervention of the central processor
Direct memory access	Yes	No

Table 2: *Overview of reviewed A/D software capabilities. Where optional packages are available, capabilities are for the most extensive set of compatible functions.*

	SALT	LABPAC	PCLAB	Labtech Notebook	ASYST	ILS
Company	Sam Fensten 4949 South Woodlawn Ave. Chicago, IL 60615	Scientific Solutions Inc. (Tecmar) 6225 Cochran Rd. Cleveland, OH 44139 (216) 349-4030	Data Translation Inc. 100 Locke Dr. Marlboro, MA 01752 (617) 481-3700	Laboratory Technologies 255 Ballardvale St. Wilmington, MA 01887 (617) 657-5400	Macmillan Software Co. 866 Third Ave. New York, NY 10022 (212) 702-3241 (800) 348-0033	Signal Technology Inc. 5951 Encina Rd. Goleta, CA 93117 (800) 235-5787
Price	$50	$495	$249	$895	$550 (one module) to $2195[1]	$995 to $2490[1]
Supports DT2801	No	No	Yes	Yes	Yes	Yes
Supports Lab Master	Yes	Yes	No	Yes	Yes	No
Language	TIL interfaced to BASIC	BASIC, Pascal, FORTRAN	BASIC, Pascal, FORTRAN	Menu-driven	Independent, FORTH-like TIL	Single instructions at a time
Memory size	64K BASIC plus two 64K integer data buffers and one 64K floating-point data buffer	128K	64K in BASIC, more in newer version using FORTRAN and Pascal	Up to 400K for data	320K for programs and data, 64K data buffers up to total memory in machine	Buffers up to total memory in machine
Digital I/O	Yes	Yes	Yes	Yes	Yes	No
File structure	BASIC or arrays transferred between disk and high memory	Same as higher-level language	Same as higher-level language	ASCII or DIF, compatible with Lotus 1-2-3 and Symphony	IBM BASIC-compatible ASCII or own binary files	FORTRAN-compatible
Documentation	Terse, not example-based	Obscure, not example-based	Satisfactory, example-based	Satisfactory	Extensive, example-based	Satisfactory, example-based
Lab Master maximum A/D rate (100-kHz board)	48 kHz	1 kHz	—	20 kHz[2] (300 Hz)	21.9 kHz[2] (1 kHz)	—
DT2801 maximum A/D rate	—	—	13.7 kHz (2801) 27.5 kHz (2801-A)	13.7 kHz (2801) 27.5 kHz (2801-A) (300 Hz)[2]	13.7 kHz (2801) 27.5 kHz (2801-A) (1 kHz)[2]	13.7 kHz (2801) 27.5 kHz (2801-A)
Lab Master maximum D/A rate (100-kHz board)	26 kHz (two channels) 33 kHz (one channel)[3]	1 kHz	—	300 Hz	14.4 kHz	—
DT2801 maximum D/A rate	—	—	14.8 kHz (2801) 29.5 kHz (2801-A)	300 Hz	10.8 kHz	16 kHz (2801) 33 kHz (2801-A)
Supports Lab Master timer/counters	Yes	No	—	No	Yes	—
Lab Master clock functions	1 MHz divisible by power of 10 to 10^4	5 kHz to 30.2 Hz	—	No	No	—
Background data acquisition	No	No	Yes	Yes	Yes	No

	SALT	LABPAC	PCLAB	Labtech Notebook	ASYST	ILS
GRAPHICS	No	Yes[4]	No	Yes[4,5]	Yes	Yes
Number of channels displayed	—	One in high resolution, three in low resolution	—	Four for each of five windows	Unlimited	One
x,y display	—	Yes	—	No	Yes	No
x,y,z display	—	No	—	No	Yes	No
Cursor positioning and reading	—	No	—	No	Yes	Yes
Graphing maxima and minima	—	No	—	No	No	Yes
Histograms	—	No	—	No	Yes	No
DATA MANIPULATION	Yes	No	No	No	Yes	Yes
Scale and offset	Yes	—	—	—	Yes	Yes
Differentiate and integrate	Yes	—	—	—	Yes	No
Moving averages	No	—	—	—	No	Yes
Maxima and minima	No	—	—	—	Yes	Yes
Filter	No	—	—	—	Yes	Yes
FFT	No	—	—	—	Yes	Yes
STATISTICS	No	No	No	No	Yes	Yes
Mean and variance	—	—	—	—	Yes	Yes
Mode	—	—	—	—	Yes	No
Standard deviation	—	—	—	—	No	Yes
Probability distribution	—	—	—	—	Yes	No
Linear and polynomial fit	—	—	—	—	Yes	Yes

1 Prices increase with additional modules for graphics, statistics, and data analysis. Base price shown includes A/D module.
2 The highest A/D rate applies only when data is being acquired in the foreground mode and the computer is not performing other tasks. The lower rate applies when data is being acquired in the background, so that the computer can perform other routines simultaneously.
3 The D/A output for two channels is substantially faster than half the maximum conversion rate for one channel because both channels are converted following a single clock pulse.
4 These packages make displays in real time as data is being collected. Labtech Notebook makes only this type of display. LABPAC will display previously stored data as well.
5 Other graphics and analyses are intended for use with Lotus 1-2-3.

Several types of data manipulations can be performed while data is being collected.

any order. They consist of a list of calls to previously defined words. Program execution is rapid because all the primitives are run in machine language, and the only extra time is that required to pass parameters to each subroutine and to proceed from one routine to the next. A well-known example of a threaded interpretive language is FORTH, and one of the software packages, ASYST, is written in a FORTH-like language that provides a great deal of speed and flexibility. It also provides a disk-file structure that can be read later by other ASYST programs for subsequent data analysis. However, there are three major disadvantages in using a TIL for the entire programming. As with compiled languages, you lose control of the program once you start it. Also, many complex commands, such as print-formatting procedures, must be custom-built as secondary words that are listed in an individual user's "dictionary," the size of which usually increases with time. This type of language has a great deal of flexibility, but it fails to include a lot of procedures that exist in more complete languages. Finally, complex conditional branching routines are difficult to follow in threaded interpretive languages.

The approach we took with SALT was to interface a TIL to BASIC. The TIL is used just for the laboratory applications, such as data acquisition and experiment control, and for data storage. BASIC is used for program control and most analyses. This approach provides the speed of the TIL for real-time applications and the interactive characteristic of BASIC for all other applications.

MEMORY LIMITS

Many forms of experiments are associated with the initial collection of a large amount of data. In some cases, the data is collected in much the same way that an oscilloscope trace is photographed. That is, one or more electronic signals are recorded in digital form for a specific period of time. To provide adequate resolution, the signals are digitized at fairly frequent intervals, so that in typical applications, a few kilobytes of data is collected for each signal. This much data would not tax the memory of computers that accommodate 512K bytes of memory, except when the entire program is run in BASIC.

The IBM PC BASIC is limited to 64K bytes of memory for program and data. Either a lengthy program or a large amount of data can easily outgrow this space. Thus, PCLAB or LABPAC run exclusively in BASIC can be inadequate in applications that require more than 64K bytes of memory, but when you encounter such limitations, you can use a compiled language such as FORTRAN or Pascal. The other programs provide considerably more space for data, as indicated in table 2.

PERFORMING MULTIPLE FUNCTIONS

As described below, the A/D routines in some of the packages will perform single functions nearly as fast as the hardware on the boards will allow. However, program execution can be greatly slowed when more than one function is to be performed at once.

For example, it is often desirable to output a control voltage, either through a D/A channel or through an I/O port, at the same time that data is being collected in the A/D channels. ILS cannot do this, because it accepts only one command at a time. It is possible to perform multiple tasks in sequence using batch files in PC-DOS, but since the routines for each command are read from disk each time they are called, time between commands is extremely long and variable. At the other extreme, routines that are called from compiled languages can pass from one routine to the next within a few tens of microseconds. The exact interval required varies according to the numbers of parameters that must be passed to the new routine, and thus we cannot give a single interval here.

When commands are called individually from BASIC, the time required to pass from one command to the next is about 2 milliseconds, so the time required to stop sampling data, output a control voltage, and resume data collection would be about 4 milliseconds. The same sequence would take about 150 microseconds in SALT and about 600 microseconds in ASYST. We have examined only the BASIC versions of LABPAC and PCLAB, which each take about 4 milliseconds for this sequence. The FORTRAN and Pascal versions should run substantially faster.

Another example of desirable simultaneous functions is dealing with recently acquired data while more data is being collected. The packages that have provisions for background data collection (see table 2) have considerably slower A/D conversion rates, but these are still sufficiently fast for many purposes.

Several types of data manipulations can be performed while data is being collected. For example, some analysis might be carried out to reduce the amount of data to be stored. Alternatively, some display of the data and/or the analysis might be made. Those packages that provide for the sequential operation of multiple commands (all but ILS and PCLAB) will allow for some form of data manipulation during data collection, although the collection rates must be relatively slow for those programs that are run entirely in BASIC. A type of manipulation that is not possible with most of the packages is the simultaneous storage of data on disk as it is being collected, termed "streaming to disk." Only one of the packages, Labtech Notebook, provides for this streaming. This can be an extremely valuable feature when large amounts of data are to be collected continuously, as with an analog strip-chart recorder. The data-collection rates are necessarily slow when streaming is being performed, but it does allow an enormous amount of data to be collected. With a standard 10-megabyte

(continued)

hard disk drive, 5 million integer samples can be collected at a rate of 300 Hz and stored for later analysis using Labtech Notebook.

A/D Conversion

Perhaps the most time-critical function of an interface board is the rate at which it can digitize analog data and put it into memory. Except for the high-resolution converters that have slow conversion speeds, the converters supplied by the manufacturers can operate at least as fast as the software packages allow.

All the packages have provisions for A/D conversion, and most will run at speeds close to the maximum speed of the standard A/D converters on the boards. A disappointing exception to this is LABPAC, which Scientific Solutions provides for its Lab Master board. The maximum timing interval that this package provides for is 1 kHz, even though the clock on the board itself runs 1000 times faster. LABPAC has a provision for making faster conversions using a separate, external clock, but the manual does not specify a maximum rate and we have not used an external clock to test the routine.

D/A Conversion

When you use the D/A converters to output voltages to control experiments, the speed of the output depends upon the required waveforms. If you require single voltage steps, great speed is not essential, and any of the packages that can operate the D/A converter will generally be sufficient for this purpose. However, when you require continuously changing waveforms, the smoothness of the waveform will depend greatly on the speed of the routines that run the converters; faster conversions let you use smaller voltage increments in providing a given total amplitude of waveform. In these applications, the faster routine will generally be more satisfactory.

Speed in D/A conversion is even more critical when the converters are used for oscilloscope displays. In general, displays must be made at rates of at least 15 Hz to avoid serious flicker. Since you will typically display as many as 1000 samples per sweep, you want a D/A converter capable of at least 15 kHz for this kind of application. Speed is not nearly so essential when you display hard copy using analog recorders, and since the computer also has the ability to display waveforms on its monitor, the use of D/A conversion for oscilloscope displays is not as essential as it might otherwise be. However, it can be useful in many laboratory applications.

Digital I/O Ports

These ports, originally designed for transferring parallel digital data, are extremely useful for transmitting individual control signals between the computer and the experimental apparatus. All experiments have some need to synchronize the A/D recording of the computer with the experiments. Since each of the 8 channels in an I/O port can transmit a separate voltage signal, 24 separate channels are available on the Lab Master and 16 come with the DT2801.

The packages that provide the ability to use these I/O channels (table 2) allow a great deal of interaction between computer and experiment. Without the ability to use these ports, you must find some other method of synchronizing the computer with the experiment. For example, it is possible to use a D/A channel to output a control voltage and to accept digital signals through the A/D channels. However, there are fewer of these channels, and to the extent that they are used for digital communication, they cannot be used for other functions.

Clock Functions

Each board has a clock, and the frequency of these clocks is divided down by powers of 2 on both boards, and by powers of 10 as well on the Lab Master (table 1). Division by powers of 2 can produce some awkward clock frequencies, so in developing SALT we have used only the powers-of-10 dividers. You can obtain frequencies other than those provided by the clock dividers by using the timer/counters on the Lab Master board, but they are not available on the DT2801. Thus, you can obtain much finer gradations in the clock frequencies with the Lab Master.

With most of the software packages reviewed, the actual operation of the clock and timer/counters is invisible. You select a trigger frequency, and the software establishes the correct timer output. In some cases, this reduces the number of timer/counters available to you when a timer/counter is not necessary for the desired frequency.

TIMER/COUNTERS

We have found these timer/counters extremely useful in high-speed operations. For example, it is possible to put out a control voltage to the experimental apparatus at precise times during data collection at the maximum rates. This is done by initializing the timer/counters to count the same pulses that are being used to trigger the A/D converters but inhibiting their operation until the series of A/D conversions is initiated. When each of the counters reaches its terminal count, the voltage output changes without any intervention from the central processor, so the operation of the timer/counters does not take up program-execution time during the critical period of data acquisition. It is also possible to generate different frequencies of trigger signals for different purposes. For example, one frequency might trigger A/D conversions, while the other triggers external events in the experiment.

Of the commercial packages that run Lab Master, only ASYST provides control of the timer/counter (table 2). However, ASYST reserves four of the timer/counters for its own operation so that only one is available to you. Our own SALT usually uses one of the counters for clocking the A/D conversions, so that when you use it in that mode, four of the five timer/counters are available to you. It is possible, however, to use the output of the clock dividers directly in SALT, so that all five timer/counters can be available.

SYNCHRONIZATION

When you program the computer to send or receive digital signals, you can synchronize its data collection with external events. The accuracy of this synchronization depends upon whether the timing clocks can be synchronized and whether the interrupts within the computer can be disabled. The time-of-day clock within the computer generates a 250-microsecond interrupt every 55 milliseconds. If this interrupt is not disabled, an uncertainty of 250 microseconds occurs in every synchronization. As nearly as we could determine, most of the packages disabled this interrupt during data collection, but PCLAB requires that you write a routine specifically for this purpose when the interrupt is to be disabled.

A less serious problem occurs when the clock is made to run continuously and an external event initiates A/D conversion. In that state, synchronization uncertainties equivalent to one clock interval occur with each conversion. The commercial packages do not provide for synchronizing the timers exactly with external events.

GRAPHICS

Like SALT, PCLAB makes no provision for graphics. Two other packages, Labtech Notebook and LABPAC, let you display the channels of digitized data as they are collected. LABPAC also allows display of previously stored data and x,y plots of two separate channels. Neither of these two packages offers any more extensive graphics.

Most of the graphics routines come as part of the basic ILS package, although some additional graphics are part of an extra-cost data-analysis module. The graphics in ASYST come as part of an extra-cost module. Data obtained with Labtech Notebook can be graphed using Lotus 1-2-3.

DATA ANALYSIS

As with the graphics, ASYST and ILS offer extensive data-analysis routines at an increased price. These two products also offer substantial statistical packages. ASYST's statistics are included in the graphics option, while in ILS the statistics are included in the data-analysis option. As with the graphics, you can perform analysis of

(continued)

Labtech Notebook was easy to learn but was either too simple or too slow.

data obtained with Labtech Notebook in Lotus 1-2-3.

DOCUMENTATION

Most of the manuals were easy to understand and gave satisfactory explanations of each of the commands in the package. The documentation with ASYST was voluminous, mainly because it had to describe a whole new language. We found that it would take longer than we were willing to spend to fully appreciate the richness of this system. One of our colleagues and his technician spent several weeks studying the manual and at the end of that time still did not feel that they had mastered the language.

At the other extreme, we quickly learned the single-command package, ILS, and the menu-driven package, Labtech Notebook. We have worked with only the BASIC version of PCLAB. We found the manual for PCLAB very helpful in its explanations of the routines and the board. It gives numerous useful examples. We had some difficulty in obtaining specific information from the LABPAC manual, which gives very few examples.

IDIOSYNCRASIES

As with all new software, some relatively minor aspects of these packages did not operate exactly as we would have expected after reading the manuals or the advertising. In many instances, the manuals were inaccurate or misleading. For example, the documentation for the early version of Labtech Notebook stated that data could be acquired in the background mode at rates of up to 600 Hz when in fact the highest possible rate was only 300 Hz. Later versions of the manual have been changed to reflect the more limited rate. This type of disparity between manual and program differs from an actual bug, in which the program does not operate correctly. An example of a relatively harmless bug in ASYST was that voltage values obtained through the A/D converters were half those expected from the input voltages and the amplifier settings. We never did know the reasons for this discrepancy.

We have refrained from cataloging these small deficiencies for two reasons. First, we did not test every aspect of every package, so such a catalog would be incomplete at best. The unwary reader might thus be given the false impression that some aspect of a particular package worked perfectly because we failed to state that it didn't. Second, for most users, the major difficulty will not be an actual bug in the program but some subtle characteristics of the routines that prevent them from carrying out a required task.

A good example is the problem described above of exactly synchronizing the initiation of A/D conversion to an external event. We once had such a problem with synchronization in the laboratory and have written the SALT A/D routines so that they can start the A/D clock after the external trigger is received. We could not find a process for such synchronization in any of the other packages, and it is not possible with the DT2801 board, which has only a free-running clock. This problem of synchronization is just one of many that you might encounter in a specialized application, and it is not possible to give a comprehensive list of the pitfalls that you might encounter in the diversity of laboratory applications. We can only caution prospective buyers of any of the packages to examine its characteristics carefully in light of individual protocols before purchasing the package.

Consequently, we have chosen to describe only a few of the major problems we found with the different packages.

We had the greatest difficulty with ASYST because it was so complex. It is undoubtedly a powerful system, but we estimate that it would take an experienced programmer at least a few months of full-time work to be able to use the routines effectively. However, the company has a user-support program that we found to be very helpful when we telephoned.

Labtech Notebook was easy to learn but suffered from the drawback that it was either too simple or too slow. When run in its fast foreground mode, it could not do anything but the laboratory interfacing tasks assigned to it. It cannot, for example, collect data for brief periods at a rapid rate and then spend some time analyzing or displaying the data before beginning another period of rapid data collection. This deficiency occurs because Labtech Notebook does not provide data-analysis and display routines of its own but relies on Lotus 1-2-3 for these functions. Lotus 1-2-3 and Labtech Notebook cannot operate in the foreground mode simultaneously. When collecting data in the slower background mode, Labtech Notebook can analyze data using any other language; but the highest conversion rate is 300 Hz.

We found LABPAC to be very limited in its functions, and we found the manual to be somewhat misleading. Scientific Solutions has a user-support system, but it took us six telephone calls to the company before we could speak to someone willing to answer our questions. Even then, the individual with whom we spoke was not able to give specific answers to some questions about the capability of the routines.

CONCLUSION

The software packages reviewed here differ in their emphasis on hardware control, data manipulation, and data analysis. At one extreme, ILS provides few experimental control functions but extensive data analysis. Other packages (PCLAB, Labtech Notebook, and ASYST) provide background data acquisition while other programs operate in the foreground. The two most general packages, ASYST and Labtech Notebook, operate both boards. Labtech Notebook is menu-driven and therefore very easy to operate. ASYST, perhaps the most extensive package, offers separate modules for analysis, acquisition, and graphics/statistics but requires a great deal of time to learn its independent language. ■