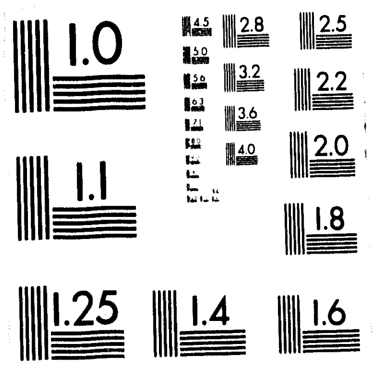

1.0
4.5
2.8
2.5
5.0
3.2
2.2
5.6
3.6
6.3
2.0
7.1
4.0
8.0
1.1
1.8
1.25
1.4
1.6

1 of 2

120
5-11-94 JS ①

NIPER Lab WARDEN

Volume 1

Description and LabVIEW® Executable Code of a General-Purpose Laboratory-Automation Program

By
S. M. Mahmood and D. K. Olsen

April 1994

Performed Under Cooperative Agreement No. DE-FC22-83FE60149

IIT Research Institute
National Institute for Petroleum and Energy Research
Bartlesville, Oklahoma

Bartlesville Project Office
U. S. DEPARTMENT OF ENERGY
Bartlesville, Oklahoma

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield VA 22161

NIPER-689 (Vol. 1)
Distribution Category UC-122

NIPER Lab WARDEN
Volume 1
Description and LabVIEW®
Executable Code of a General-Purpose
Laboratory-Automation Program

By
S. M. Mahmood and D. K. Olsen

April 1994

Prepared for
U. S. Department of Energy
Under Cooperative Agreement No. DE-FC22-83FE60149

Thomas B. Reid, Project Manger
U. S. Department of Energy
Bartlesville Project Office
P.O. Box 1398
Bartlesville, Oklahoma 74005

Prepared by
IIT Research Institute
National Institute for Petroleum and Energy Research
P.O. Box 2128
Bartlesville, OK 74005

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

TABLE OF CONTENTS

TABLES

ILLUSTRATIONS

TABLE OF CONTENTS—Continued

TABLE OF CONTENTS—Continued

TABLES

TABLES

ILLUSTRATIONS

TABLE OF CONTENTS—Continued

TABLES

TABLE OF CONTENTS—Continued

ILLUSTRATIONS

NIPER Lab WARDEN

Volume 1
The Description and LabVIEW® Executable Code of a General Purpose Laboratory Automation Program

by S. M. Mahmood and D. K. Olsen

ABSTRACT

This report is Volume 1 (the description) of a two volume series that describes a general purpose, automation computer program developed by NIPER for data acquisition/control/analysis/presentation. This software was developed to provide interactive computer control of a variety of instruments typically found in laboratories and pilot plants in order to improve efficiency in operation and safe handling of potentially hazardous operations. For example, it is easily adaptable for operating a laboratory that conducts experiments at extreme conditions of pressure and temperature, such as those found in a steamflooding laboratory. The software was developed in an object-oriented graphical language around National Instruments' LabVIEW® which is the future trend in automation programming.

EXECUTIVE SUMMARY

Volume 1 (the description, this report) and Volume 2 (panels and diagrams for NIPER Lab WARDEN) describe a computer program that occupies two 1.4 Meg floppy disks (auto extracting compressed file) that are part of the software originally developed to operate NIPER's thermal oil production research laboratory. This is a general program that can be readily adapted by other users to their specific laboratory or pilot plant application. The use of an object-oriented graphical (symbolic) computer language in NIPER's programs permits easier and faster adaptation than the typical line-code languages such as FORTRAN. The programs look like logic flow diagrams (electrical circuitry) rather than typical syntax line codes. They offer significant advantages over custom-made programs when flexibility and/or networking is desired, since it is easy to reconfigure (during run-time) and allows interaction between several remote locations running on this program. Technology transfer is our goal in releasing this software, to help smaller independent facilities which are looking for an integrated automation program, but cannot afford a sophisticated commercial program. NIPER's program is very likely to provide enhancements over custom-made programs in situations where instrument requirements change frequently, or when several groups need to interact or share data over a network. One of the advantages of a uniform, standardized, and flexible automation program is that information/experience/training can be shared to increase efficiency, safety, and reliability. More specifically, NIPER's program provides the following features:

(1) allows the user to select any number of instruments and specify their parameters for a particular run. The selection can be made during run-time, or one of the previous selections can be chosen prior to a run. The devices/instruments can be added/deleted/reset anytime during a run;

(2) acts as a liaison between instrument-related software (driver programs) for various instruments, i.e., links up or connects various driver programs in order to bring about proper coordination of activities;

(3) provides means to monitor instruments so that the user is informed if values go out of user-specified range;

(4) eliminates operator errors by setting constraints;

(5) allows the user to automatically rectify minor problems by specifying what to do if an instrument's value falls into a warning range, or if there is an abrupt change, i.e., automatically reset one or more devices to a specified percentage of their current value;

(6) allows the user to automatically handle emergencies by specifying an emergency response sequence so that if an instrument(s) value(s) fall outside the acceptable range, or if there is an abrupt change, the devices can be automatically reset to safe values in an orderly fashion;

(7) provides easy access and editing to current and previous data with advanced graphic features; and

(8) provides a visual display of the status of a test in progress for quick trouble-shooting.

OBJECTIVE

The objective of this report is to provide documentation of a software program that was primarily developed for NIPER's thermal laboratory, but is easily adaptable for use by other laboratories. The thermal laboratory is designed to evaluate the merit of various thermal methodologies that show potential for recovering more oil from porous media. The experiments involve injection of steam and additives (surfactants, gases, solvents, and other chemicals) at specific operating conditions to simulate petroleum reservoirs on a small scale. NIPER's thermal laboratory is designed to run at high temperature (550° F) and high pressure (1,000 psi) with steam and flammable light-hydrocarbons. Under these conditions, safety is an important consideration. Operating these experiments manually was prone to error with potentially serious consequences; therefore, automation was needed to conduct experiments safely. Automation was expected to improve laboratory efficiency, data quality, reduce analysis time as well as obtain data accurately and efficiently from remote location.

The general purpose software developed at NIPER is a user-friendly and easily modifiable program suitable for a variety of laboratories and pilot plants for their data acquisition/control/analysis/presentation needs. This software will provide laboratories or pilot plants of moderate complexity: (1) a simple yet powerful automation program that is inexpensive, flexible, easy to use/modify, and capable of monitoring test progress and taking corrective actions automatically when needed; (2) an effective program that does not require a workstation for effective utilization; (3) a program that allows interaction with other major software packages to enhance its capabilities, e.g., to automatically open Microsoft Excel, transfer data in its preferred format, and plot this data; thus, saving time and effort over doing it manually; and (4) a program that allows real-time access to the input and output data so that system set-up can be changed and reports can be generated in real-time.

FORMAT OF REPORT

Chapter 1 of this report introduces LabVIEW®—the language used to develop this software—and presents the concepts behind object-oriented and graphical languages. Chapter 2 provides an overview and a conceptual framework of NIPER's automation programs. The following six chapters (3-8) describe the software program (facility) as follows: Chapters 3, 5, and 7 provide general description of the three main facilities: NIPER MAIN FACILITY, NIPER Lab WARDEN, and NIPER GRAPHIC FACILITY, respectively. Chapters 4, 6, and 8 describe how to use these facilities. Chapter 9 includes example applications/problems. Volume 2 (the companion volume) contains the complete program—the hierarchical structure, explanation of the structure, and the panels and diagrams for the program code in an organized structure containing additional documentation. The complete set of panels and supporting diagrams can be printed from the programs in the floppy disks, but printing may require several days because of the large size of the files and the length of time required for printing on most laser printers.

Individuals have different levels of proficiency in the use of computer operating systems, object-oriented programming (LabVIEW), and communication hardware (plug-in boards, etc.). The authors have assumed two different levels of users. On one level are operators who will take the program and adapt the current configuration to their needs. On another level are programmers who may wish to extensively revise and develop their own program around the NIPER programs provided. This second type of user is assumed to be more computer literate and knowledgeable about object-oriented programming structure. With regard to Macintosh computers, the user is assumed to have a good understanding of the fundamentals of System 7® operations (Apple®, 1991). The second level user should also have basic proficiency in

LabVIEW programming obtained by going through the training exercises (learn by doing), which are found in the National Instruments LabVIEW literature (National Instruments Corp., 1991).

For automation programs (laboratory, pilot plant, or production facilities) dealing with instrument control, it is imperative that the validity and/or reliability of the program be established with reasonable certainty. No warranty or guarantee of the applicability of the programs is implied by NIPER or DOE. The user must ascertain the suitability of the system for the user's own specific needs.

ADVANTAGES OF OBJECT-ORIENTED AND GRAPHICAL PROGRAMMING APPROACH

Object-oriented graphical programming is an approach that has received great attention and acceptance in recent years. Since Macintosh® and Microsoft Windows® for PC operating systems became available, most commercial software has been developed using the object-oriented approach. The same trend is also seen in lab automation programming, where object-oriented graphical languages are replacing line-code programs as the language of choice.

In object-oriented graphical programming, the program is coded as a series of symbols (objects). The related data for each object are packaged along with the program code. Objects can be defined within objects. This hierarchical structure allows the breaking down of complex programs into smaller objects. Since data are packaged along with the code for each object, their present value is saved each time the program is run and saved.

There are many advantages of object-oriented graphical languages. First, writing, debugging, and modifying the code are easier because each object can be treated as an independent program not linked to other programs (each object interacts with other objects only by sending and receiving messages). Second, the objects and their input/output data can be pictorially depicted by icons which increases the clarity for both programmers and users. Third, several processes or programs can be run simultaneously by considering each program as an object. By merely switching their active status, they can run intermittently and interact with each other. This also extends a programmer's capabilities to use external codes. The linking of multiple processes and objects increases user-friendliness and error-handling capabilities.

TECHNICAL REQUIREMENTS

NIPER's automation software was written on an Apple Macintosh II computer using LabVIEW®, an object-oriented language from National Instruments. The user must acquire a licensed version of National Instruments' LabVIEW software to legally execute the NIPER program. LabVIEW was selected as an operating platform because it allows these programs to be run on Macintosh, IBM-compatibles, or Sun workstations with an appropriate version of

LabVIEW software. LabVIEW 2.0 requires 4 megabytes of RAM and >16 megabytes of hard disk space. The NIPER programs, along with necessary LabVIEW software, require 6 megabytes of RAM and 6 megabytes of hard disk space. If supporting software such as Microsoft Excel®, Microsoft Word®, or other graphics software such as Spyglass® are to be interacted with NIPER programs, their memory requirement needs to be added. However, it is not necessary to run every NIPER program simultaneously; thus, the program can be run with 2-3 megabytes of memory at a minimal level.

Users do need to provide and load the driver programs for communicating with their unique instruments. A driver program is a mediator/translator between the computer and an instrument. Some instruments may communicate in unique command languages. The LabVIEW package includes several driver programs, but many more standard driver programs are available from third party consulting/software development companies. A template and examples of driver programs are included in NIPER's package to ease the task of writing a driver program. A driver program can be written or modified by experts, on the average, in a few hours. National Instruments also provides excellent technical support by telephone, fax, or mail. All above options require at least a cursory knowledge of LabVIEW. Inexperienced users should seek help from others in obtaining driver programs.

PREPARATION OF THIS MANUAL

This report (manual) was compiled in Microsoft Word®, and figures were drawn in Canvas® 3.0, Microsoft Excel® and LabVIEW® 2.2. Some panels and wiring schematics were also pasted into the Word text by using the screen capture program Screen Shot®. Clarity and color coding details of the figures were sometimes lost in converting to a black and white copy.

ACKNOWLEDGMENTS

This work was sponsored by the U.S. Department of Energy under cooperative agreement DE-FC22-83FE60149 as part of project BE11A, Thermal Recovery of Light Oil. The authors wish to thank A. Strycker, B. Linville, and M. K. Tham, all of NIPER; and T. B. Reid of the DOE Bartlesville Project Office for their critical reviews and the staff of National Instruments for their review of a preliminary draft. The authors thank Johnathan Grigsby, Rebecca Horning, Amee Kendall, Danny Lemmons, William Lucas, Gautam Sharma, and Yesh Tyagi, AWU Summer Interns, for their assistance in testing the parts of the software, preparation of the schematics for this manual, and for testing assigned problems (software and hardware) in chapter 9.

BIBLIOGRAPHY

Apple Computer Inc., 1986. *Macintosh II User Manual.* Cupertino, CA.

Apple Computer Inc., 1986. *Inside Macintosh, Volume VI.* Cupertino, CA.

Dijkstra, E. W. Notes on Structured Programming. Contained in Reference 10, pp. 1-82.

Dahl, O. J., E. W. Dijkstra, and C. A. R. Hoare, 1972. *Structured Programming.* Academic Press, London.

Kirkman, I. W. and P. A. Buksh, 1992. *Data Acquisition and Control Using National Instruments' "LabVIEW" Software.* Rev. Sci. Instrum., v. 63, No. 1, January, pp. 869-872.

Kodosky, J., J. MacCrisken, and G. Rymar, 1991. *Visual Programming Using Structured Dataflow.* Proceedings of the 1991 IEEE (Institute of Electrical and Electronics Engineers) Workshop on Visual Languages, Kobe, Japan, Oct. 8-11. Reprinted by IEEE Computer Society, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1264.

Liles, K., 1991. *Data Acquisition Software Automates Automotive Fuel Injector Test Facility.* Engineering & Management, October/November, pp. 18-21.

Liles, K., 1992. *Data Acquisition—A Mac-Based System for Fuel and Lubricant Testing.* Scientific Computing & Automation, January, pp. 19-23.

National Instruments Corp., 1991. *LabVIEW 2—Getting Started Manual,* Part No. 320246-01. Austin, TX, April.

National Instruments Corp., 1991. *LabVIEW 2—User Manual,* Part No. 320244-01. Austin, TX, September.

National Instruments Corp., 1991. *Training In-depth course on LabVIEW 2,* Version 1.4, Part No. 776393-01, Austin, TX, April.

National Instruments Corp., 1991. LabVIEW® 2 Lab Driver® VI Library Reference Manual, Part No. 320249-01, Austin, TX, February.

Olsen, D. K., S. M. Mahmood, P. S. Sarathi and E. B. Ramzel, 1992. *Operating Guide and Specifications for NIPER Steamflood Laboratory.* DOE Report NIPER-603, August.

Olsen, D. K., S. M. Mahmood, P. S. Sarathi and E. B. Ramzel, 1990. *Thermal Processes for Light Oil Recovery.* DOE Report NIPER-515, December, pp. 18-20.

Sethi, R., 1989. *Programming Languages: Concepts and Constructs.* AT&T Bell Laboratories, Murray Hill, NJ, Addison-Wesley, Reading, MA.

Stroustrup, B., 1986. *The C++ Programming Language.* Addison-Wesley, Reading, MA.

Chapter 1

DESCRIPTION OF LabVIEW

BRIEF INTRODUCTION OF LabVIEW

LabVIEW® is a visual programming environment that can be effectively used by a broad range of people with different levels of programming skills. The two-dimensional graphical notations that LabVIEW uses are much easier to comprehend than textual notations in line-code languages. Programs look like a dataflow diagram, as shown in Fig. 1.1. Elements are pictorially represented, and data flow between elements is shown through color-coded wire-connections (lines). With this representation, constructing or understanding a program is easier and faster (National Instruments Corp., LabVIEW 2—*Getting Started Manual, 1991a, and LabVIEW 2-Users Manual, 1991c*). Complex process flow diagrams of large plant operations can be broken down into small logical units and then recombined. Thus, complex operations and their interdependence can be understood with relative ease. Kodosky, MacCrisken and Rymar (1991) have described some of the programming structure behind LabVIEW, and a number of their examples have been used in this section because of their clarity.

A simplified version of Fig. 1.1 is shown schematically in Fig. 1.2 and explained in detail under "Programming Structure of LabVIEW." Input can be an electrical signal (voltage) from an instrument, an assigned value from the front panel (which can be numeric, a string, or a series of alphanumeric characters), a string (series of alphanumeric characters), or a calculated value. A node is a process (addition, subtraction, integration, listing, sorting, etc.). An arc is the connecting link between operations, i.e., it shows where an input value leads to. A data token is an imaginary symbol superimposed on arcs to indicate that the data packet has arrived. Data tokens are used to show the execution sequence of various nodes in the program. An output terminal may be a display on the front panel, an instrument, valve, alarm, or may involve printing a file, etc.

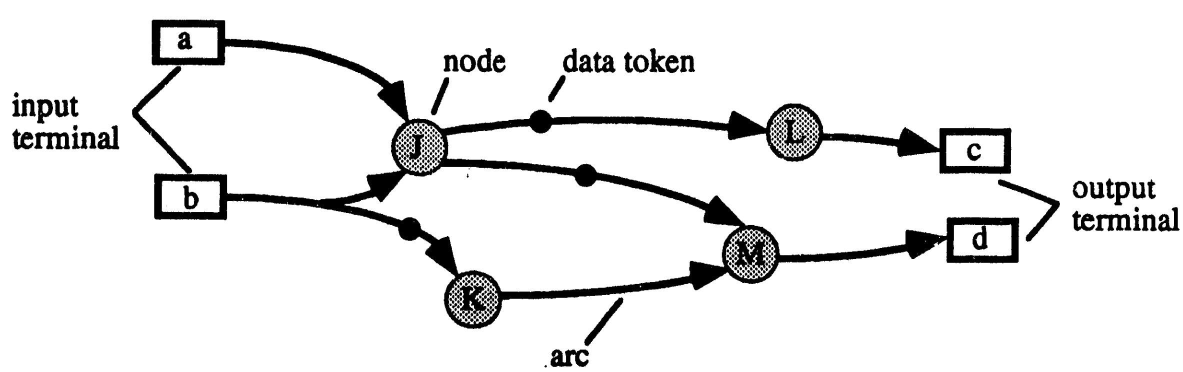


FIGURE 1.1 - A simple dataflow diagram.

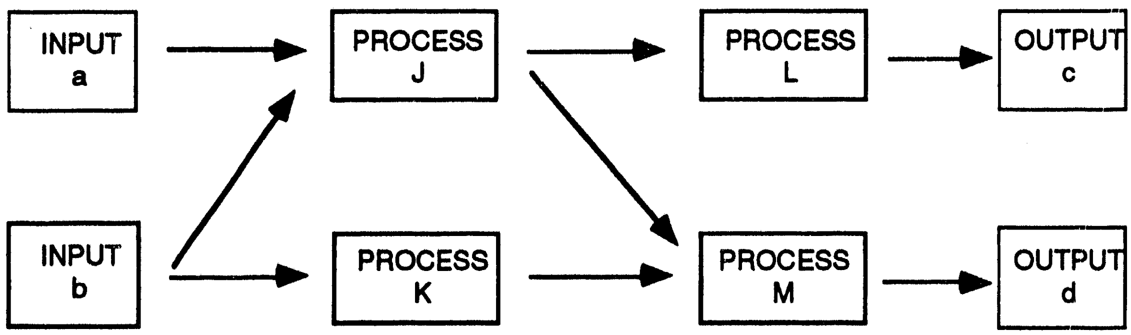


FIGURE 1.2 - Dataflow diagram simplified.

PROGRAMMING CONCEPTS RELEVANT TO LabVIEW

This section describes the concepts of programming languages that are pertinent to LabVIEW. The following discussion is aimed at highlighting the basic differences between the conventional languages and LabVIEW.

A Sample of LabVIEW Program Algorithm

Consider a simple algorithm that iteratively reads data from a file, converts it to °F, and warns the user (via an audible beep) if the value exceeds a predetermined limit. Figure 1.3 lists a FORTRAN program to achieve this objective. To help understand the structure of the program, a control-flow diagram is customarily used, such as the one shown in Fig. 1.4.

A program in LabVIEW is shown in Fig. 1.5 which achieves the same objective as the FORTRAN program shown in Fig 1.3. Notice that the program LabVIEW itself is in the form of a flow diagram, so no control flow diagram is needed. Further explanation of this program is provided in Table 1.1.

At first glance, it may appear that the FORTRAN program is easier to follow. This may be true for small programs like the one presented in this example. However, experience in writing and maintaining software shows that as the size and the complexity of the program increases, line-code programs become progressively more difficult to follow than visual programs.

Object-Oriented Programming (OOP)

The important concepts of object-oriented programming (OOP) are described below. This description is included to help readers visualize the significant advantages that this approach to programming offers over familiar line code languages such as Basic, FORTRAN, Pascal, and C. National Instrument's LabVIEW, the basis of NIPER's automation software, is a programming platform (compiler) that uses a high-level object-oriented language called "G."

```
          OPEN (5, FILE = 'ALLIANCE:LABVIEW 2:INPUT TEMPERATURE',
     +    STATUS = 'OLD') CELSIUS
C         ALLIANCE IS THE VOLUME I.E., HARD DISK, LABVIEW2 IS THE
C         FOLDER CONTAINING FILE NAMED INPUT TEMPERATURE
   10     READ (5, *, END = 99)
          FAHRENHEIT = ((CELSIUS * 9/5) + 32)
          IF (FAHRENHEIT.GT.450) GOTO 20
          GOTO 10
   20     CALL BEEP (1)
C         BEEP IS A SUBROUTINE THAT ACTIVATES AN AUDIBLE BEEP
   99     STOP
          END
```

FIGURE 1.3 - A FORTRAN program that iteratively reads data from a file, converts it to °F, and beeps if the value is higher than 450° F.

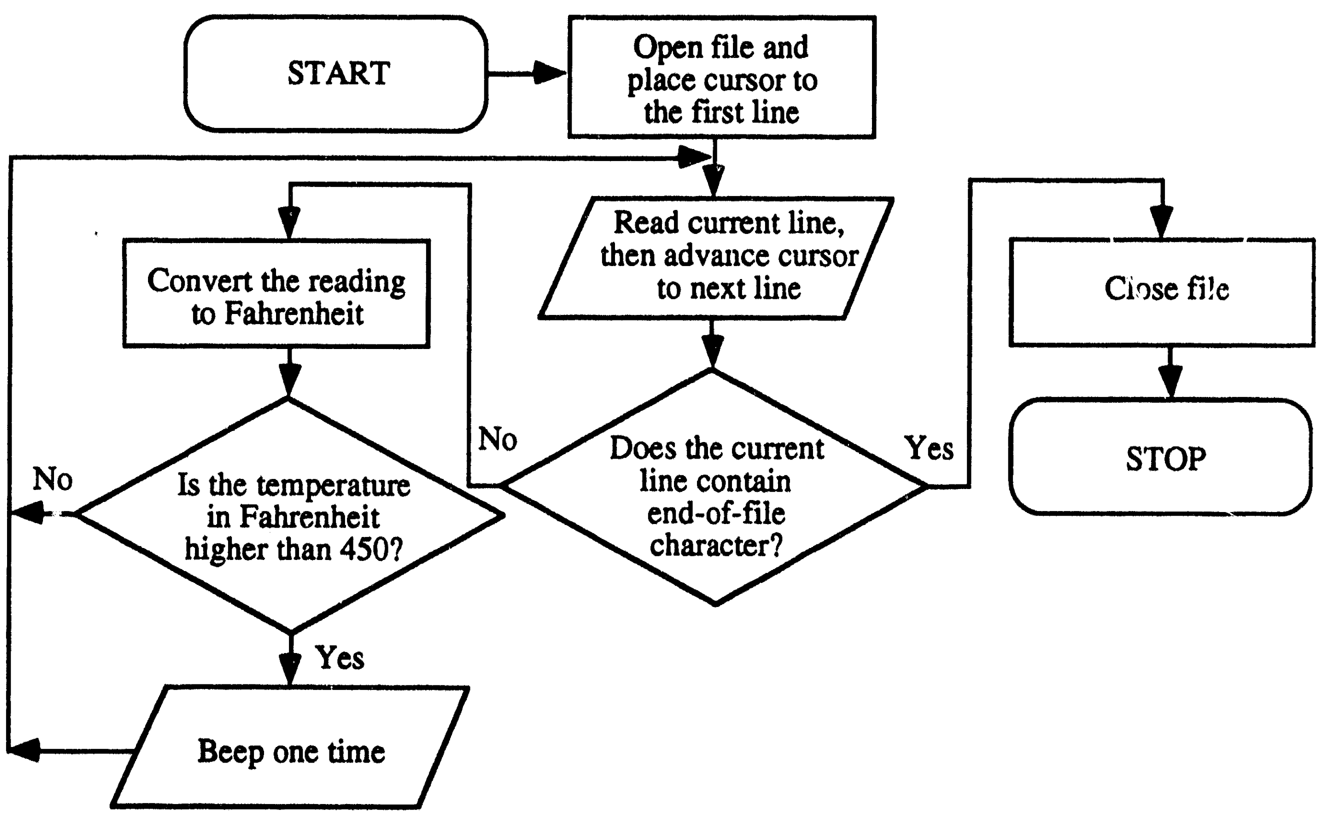


FIGURE 1.4 - The control-flow diagram of the example FORTRAN program shown in Fig. 1.3 that iteratively reads data from a file, converts it to F, and beeps if the value is higher than 450° F.

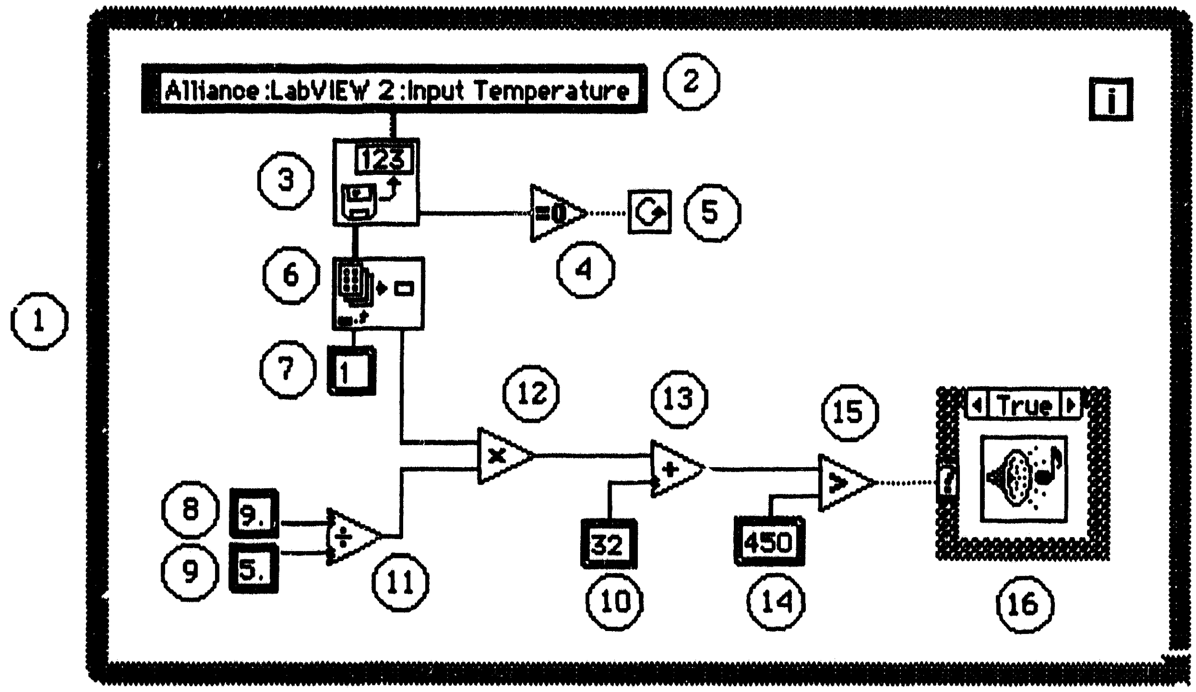


FIGURE 1.5 - A sample program segment in LabVIEW that iteratively reads data from a file, converts it to °F, and beeps if the value is higher than 450° F (see Table 1.1 for legend).

TABLE 1.1

LEGEND FOR FIGURE 1.5

1. This entire box is a "While Loop" structure. The "While Loop", which can be thought of as a single node itself, executes the diagram inside the box until the Boolean (True-False) value passed to the conditional terminal (box 5) is False.
2. This box contains the string which specifies the volume name, directory name, and file name from which the temperature data (in Celsius) is to be read.
3. This node opens the specified file and reads the current line of temperature data into a numeric array. The node then closes the file, passes the array to node 6, and passes a numeric error message value to node 4.
4. This node checks to see if the error message from node 3 is equal to zero (indicating no error). If so, the node outputs a Boolean value of True. If the error message does not equal zero, the Boolean output is False.
5. This box is the conditional terminal for the entire "While Loop". The terminal is checked at the end of each iteration and exits the "While Loop" box once the Boolean value from node 4 is False.
6. This node reads a specified element of the numeric array from node 3 and then passes the value of that element to node 12.
7. This box contains the integer value that specifies which element of the array node 6 is to read.
8. This box contains a numeric constant used in the temperature conversion from Celsius to Fahrenheit. The value is passed to node 11.
9. This box contains a numeric constant used in the temperature conversion from Celsius to Fahrenheit. The value is passed to node 11.
10. This box contains a numeric constant used in the temperature conversion from Celsius to Fahrenheit. The value is passed to node 13.
11. This node divides the value from box 8 by the value from box 9. The numeric result (a conversion factor of 9/5) is then passed on to node 12.
12. This node multiplies the output value from node 6 by the output value from node 11. The numeric result is passed on to node 13.

TABLE 1.1—Continued
LEGEND FOR FIGURE 1.5

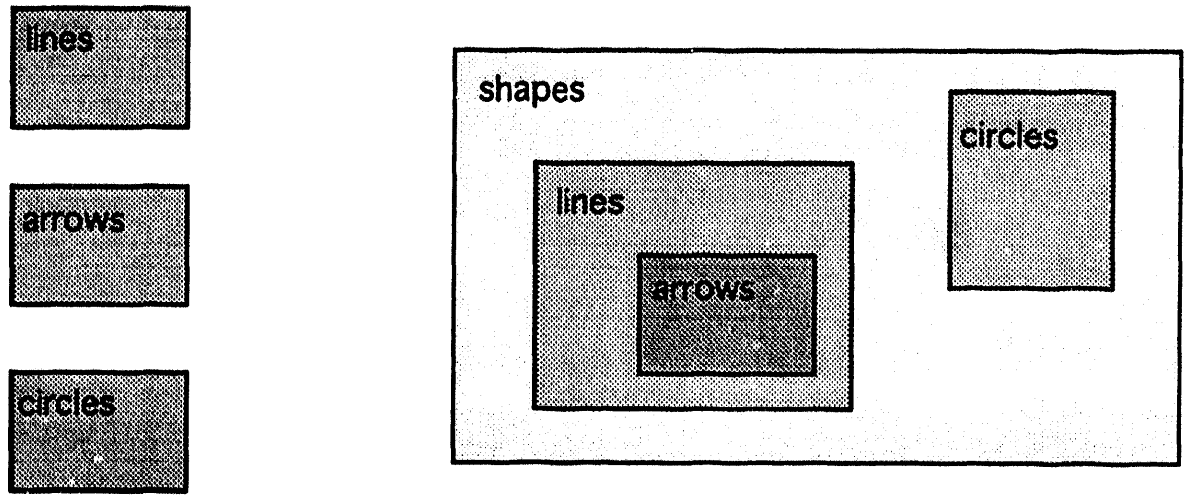
13. This node adds the value from box 10 to the output value from node 11. The resulting numeric output (Fahrenheit temperature) is passed on to node 15.
14. This box contains a constant numeric value specified as the temperature limit.
15. This node checks to see if the output value from node 13 is greater than the temperature limit of box 14. If so, then the resulting Boolean output is True. If the value from node 13 is less than the temperature limit of box 14, then the Boolean output is False.
16. This structure beeps if the Boolean output from node 15 is True, which means that the temperature value read from the file is higher than the temperature limit specified by box 14.

OOP refers to a programming style that relies on the concepts of inheritance and data encapsulation. Inheritance is a language facility for defining a new class of objects as an extension of previously defined classes. The new class inherits the variables and operations of the previous classes. Inheritance helps in building complex structures by using the existing simpler objects. Since common properties of objects can be preprogrammed by defining classes, programming effort can be significantly reduced. Data encapsulation (also called implementation hiding, meaning certain details of implementation code are deliberately hidden from the user) allows objects to be packaged so that unnecessary details of implementation are not visible from outside the object. An object may include a set of functions, procedures, subroutines, data, type-definitions, arithmetic and/or other operations. Any or all entries in an object may be defined as either public, private, or protected, depending upon their intended use. Objects can only interact with each other by sending and receiving messages.

The properties of OOP allow one module (an independent program segment like a subroutine) to be written with little knowledge of the code in another module. Modules can be reassembled and replaced without reassembling the whole system. OOP's programming style can be practiced with widely differing languages. For example, C++ (a line code language) allows both inheritance and data encapsulation to deal with the most demanding systems' tasks yet retains C (also a line code language) as a subset for tasks requiring low-level programming.[7] LabVIEW is another language that provides a simple yet powerful visual programming environment. It superimposes a graphical editing and execution system upon the object-oriented "G" language to create a platform for the users wherein the modules can be built by copying the objects from the LabVIEW library and user's own library of modules.

Procedure-Oriented Programming (POP) Versus Object-Oriented Programming (OOP)

Figure 1.6 illustrates the difference between procedure-oriented programming (POP) and OOP (Sethi, 1989). The requirement in this example is to build figures, which are basically



(a) Disjoint sets of objects (b) Nested sets of objects

FIGURE 1.6 - Different ways of classification of objects: (a) POP (b) OOP.

composed of basic shapes—lines, rectangles and circles. POP will structure a program around the operations on shapes. This may include operations for drawing, rotating, and scaling a figure. For each shape in the figure, the procedure will classify the shape and then execute the code that is appropriate for drawing that kind of shape. However, the code for each shape will contain only very elementary operations. Because of the use of elementary operations, the code for manipulating shapes is spread across the various procedures, and this is often problematic. If a new shape is added, e.g., an arrow head, then the code for handling the new shape has to be added to each procedure. Even if the new information is small, it is spread across procedures, each of which must be analyzed before the new code is added to ensure that there are no conflicting directions or assignments.

The OOP approach of handling this problem is different: A class named "shapes" is defined, which has subclasses of lines and circles. Class shapes then collects common properties, such as the height, the width, the position, and an operation for moving the shape. Properties that are specific to lines, rectangles, and circles appear in the appropriate subclasses. Inheritance, an OOP property, allows arrows to be added by extending the subclass "lines," without touching the code for the other objects. An arrow inherits all the properties of a line, so the only additional code needed to draw an arrow is the code for drawing an arrowhead. Another property of OOP—data encapsulation—allows the drawing of various shapes by simply sending messages to the class shape, without the need to see its implementation details. In OOP, each module is a completely executable program in its entirety, and it does not interact with other program segments in any way except by receiving and sending messages. This message-passing mechanism is superior to the use of "subroutine calls" in traditional programming because it eliminates any chances of inadvertently altering the data in the calling program. The user does not need to know the implementation details of a module to use it in any other module. They merely need to know the abstract information about its actions and about the input/output data to be exchanged through messages.

Since OOP is more of a philosophy (methodology) than a specific language, differentiating OOP from other programming styles is difficult to do with precision. In abstract terms, OOP relies heavily on making and using objects (building blocks) instead of using elementary units and operations. If the objects are appropriately defined, the task of manipulating them becomes easier. The objects are treated as complete units; hence, operations on them are far less likely to have inadvertent side effects, i.e., unwanted influences on other objects or programs are avoided. The power of OOP becomes more apparent as the size and complexity of a program increases. It is easier and safer in OOP to extend classes, and the code for each individual operation is relatively small, often to simply "pass the buck" by invoking operations in other objects (Sethi, 1989).

Programming Structure of LabVIEW

National Instruments LabVIEW, by using the high-level, object-oriented language "G," allows LabVIEW programs to be highly structured and modular. Each module (also called "VI" by National Instruments) is a totally independent, interactively executable program which can be used as a subprogram by other modules. To use a module as a subprogram, its icon is copied into the program. Data exchange (input/output) with the module can be accomplished by making wired-connections between the terminals shown on the icon and other elements in the program. Large programs can be developed in an hierarchical manner by starting with small modules and using them within other modules. Since each module has its own independent program and a separate input/output interface, debugging or modifying a large program becomes easier.

LabVIEW handles the execution sequence of a program in a different manner than line-code languages. In traditional line-code languages such as Basic, FORTRAN, C, or Pascal, the execution of instructions takes place according to the control-flow diagram designed by the programmer. LabVIEW, on the other hand, is based on a modified dataflow model such that the sequence of execution need not be predefined.

Dataflow diagrams specify the data dependency between computations, but they do not specifically force any particular sequence of independent computations. The dataflow diagram of Fig. 1.1 is an example. It is a directed, acyclic graph consisting of nodes, arcs, terminals, and data tokens. Terminals are the connections to the external world, and act as the sources, or sinks, of data tokens. Arcs are the directed paths over which data tokens move, and nodes are the locations in which computations are performed. A node consumes tokens on its input arcs and produces new tokens on its output arcs. What makes the diagram data-driven is the *firing rule*, which states that a node cannot execute until all of its input arcs have a data token available, at which time the node consumes one token from each input arc, performs the computation, and produces one token for each output arc. In Fig. 1.1, for example, node J has already executed, K and L are eligible to execute, and M is still ineligible because it needs a token on its second input.

In contrast to the control-flow model, the dataflow model has no concept of locus-of-control, no program counter (i.e., no sequence numbers like in text-based program codes), and no global variables (globally accessible memory). A data token exists only from its production by a node or input terminal to its consumption by another node or output terminal. All nodes that are eligible to execute can do so in any order or even in parallel; the results of the diagram will be the same in all cases.

The classical dataflow model, however, lacks the provisions for conditional or iterative computations. LabVIEW provides an extension to overcome this limitation. This extension not only preserves its firing rules and acyclic structure (thus, preserving program clarity), but also incorporates the proven benefits of the structured programming methodology. This extension involves redefining a node to be any program segment enclosed in a box-like structure that separates the body (or inside) of the structure from the rest of the program. Because the box behaves like a node as far as the rest of the program is concerned, the overall dataflow methodology is preserved. The body of a node (inside the box structure) behaves like an isolated diagram, in which access to the code is only from the top (or beginning). The program structure-semantics such as loop behavior or conditional behavior have been superimposed on the body of the box. This can be thought of as a macro structure (program as a whole) containing some micro structures (program segments inside the node), both of which independently follow the dataflow model. The micro structures, however, have some additional control properties.

Using the extended dataflow strategy, LabVIEW 2 is able to retain the important benefits of both structured programming and dataflow strategy. Furthermore, the performance of the executable code generated by compiler is comparable to that produced by a C or Pascal compiler.

Basic Facilities in LabVIEW

Editing

LabVIEW contains three interrelated editors, one for each of the three parts of a module. These three editors consist of the following: block diagram, front panel, and icon/connector. The front panel is the means of controlling the VI and interacting with the program. It specifies the inputs and outputs of the VI and is analogous to a front panel of a real instrument or an operators panel controlling a cluster of instruments. The data input terminals are called controls, whereas data display terminals are called indicators. The front panel may look simple, but it could possibly involve significant programming to support the panel functions. An example panel is shown in Fig. 1.7, which is a rendition of the "NIPER MAIN FACILITY" panel of the NIPER laboratory automation program (some of the features removed for clarity). Each front panel has an accompanying block diagram. The block diagram is the actual program (source code) created by the user with LabVIEW. An example of a block diagram is Fig. 1.8 (this is the accompanying block diagram of the panel shown in Fig. 1.7) with one of the icon/connectors expanded with its panel and block diagram icon to show the hierarchical structure. The icon/connector is a means of turning a VI into an object that can be used in the block diagrams of other VIs as if it were a subroutine. The icon graphically represents the VI in the block diagram of other VIs. The connector terminals on the icon of a VI determine where the inputs and outputs must be wired (connected) on the icon. The terminals are analogous to parameters of a subroutine.

The block diagram is a directed acyclic graph containing nodes, interconnecting wires, and source/sink terminals corresponding to the panel controls and indicators, respectively. It is constructed by copying built-in functions, structures, and previously constructed VIs, arranging them in the block diagram window, and wiring them as needed. The front panel contains all of

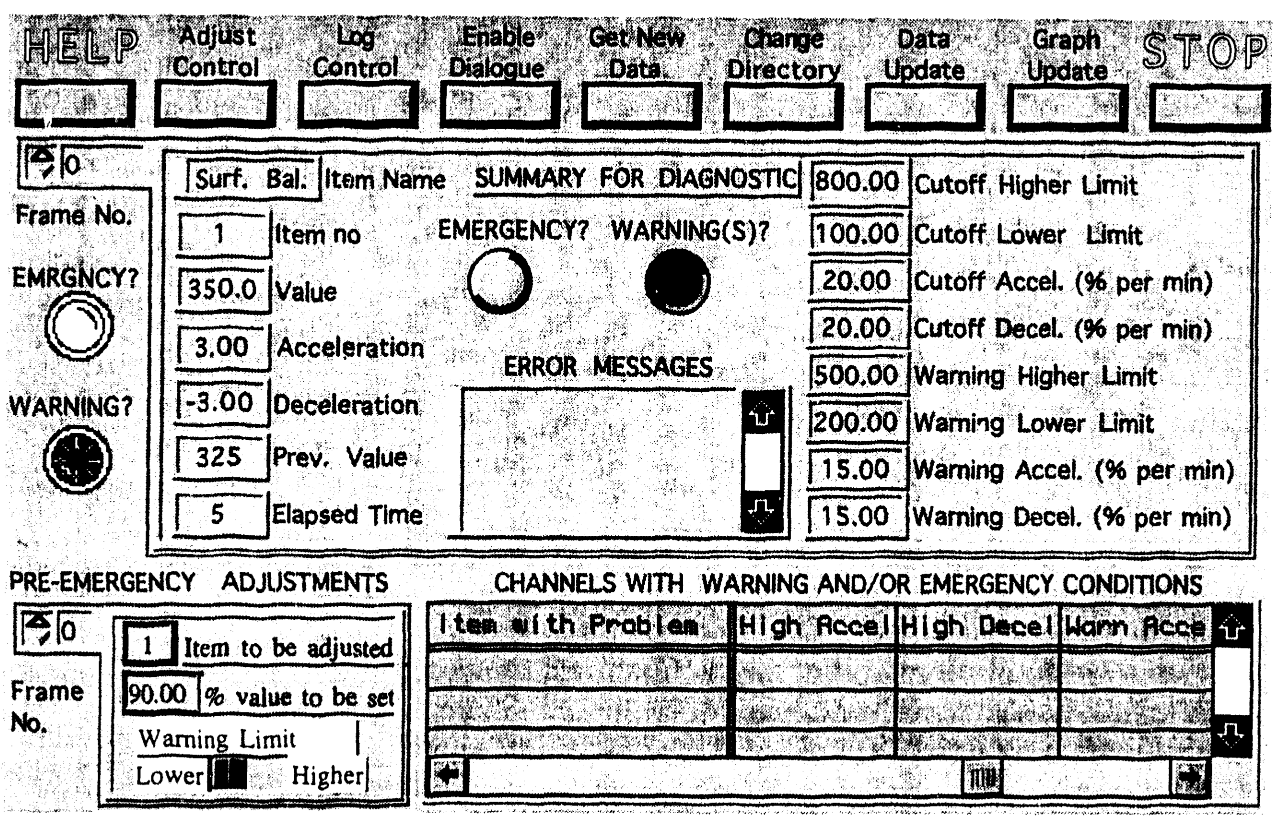


FIGURE 1.7 - NIPER MAIN FACILITY panel.

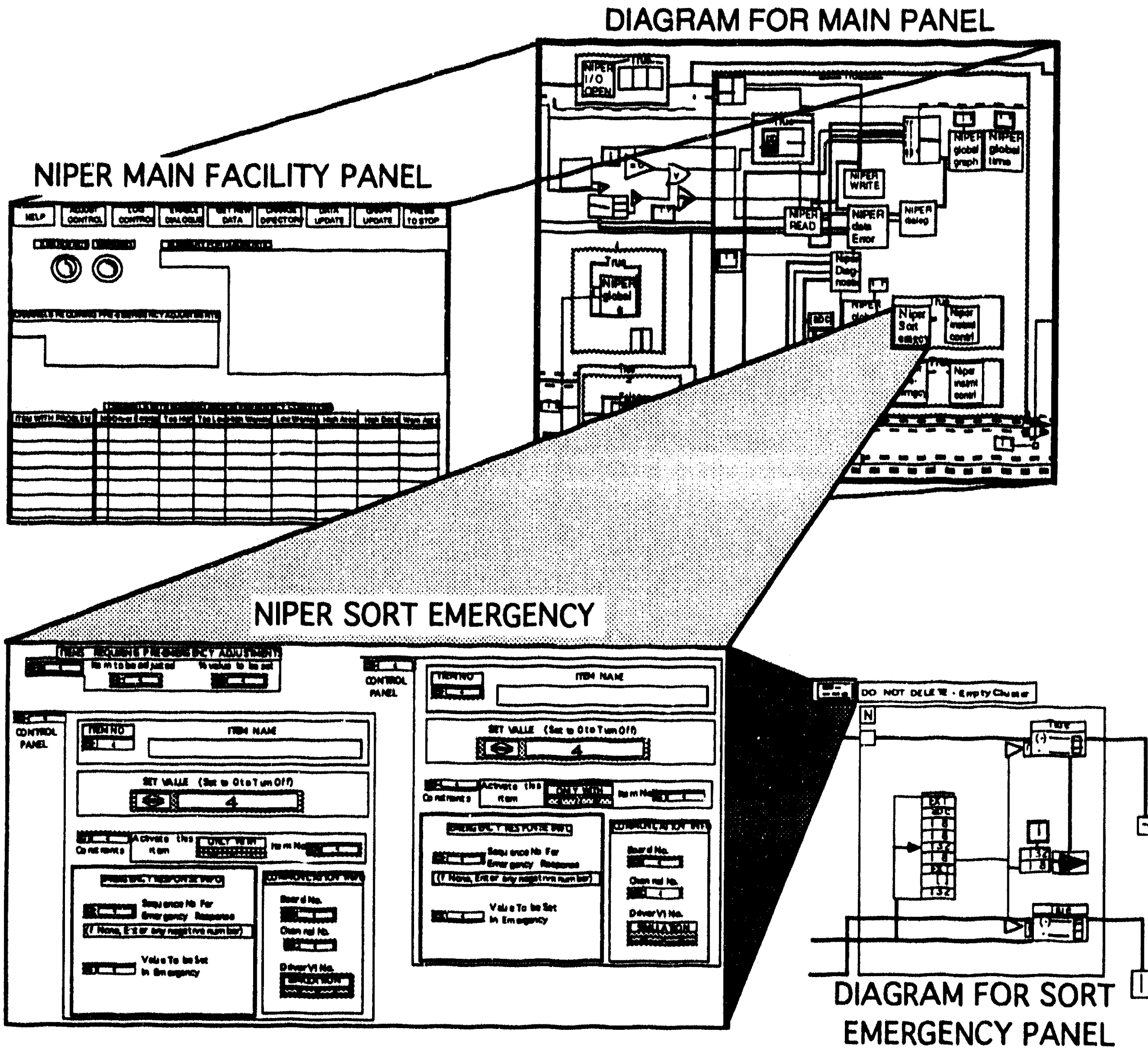


FIGURE 1.8 - Descriptive hierarchy of programming for the NIPER MAIN FACILITY showing panels, diagrams, and icons for select portions of the program.

the input/output terminals for interactive programming. These terminals define the data types of the inputs and outputs of the VI and are initialized by the previously stored user-defined values each time the VI is loaded. The icon contains terminals (non-overlapping sub regions) that are in one-to-one correspondence with a subset of the panel controls and indicators. Editing of the program without reprogramming is accomplished by activating various VI's in the program and changing the parameter values to configure the program for the user's specific application.

The wiring is done using a wiring tool to establish the paths of data exchange. The wiring tool is a cursor that looks like a spool of wire. As each edit transaction is performed, the syntax checker detects and flags any cycles introduced into the dataflow diagram, propagates data attributes (type) to all the terminals, computes the data type for each built-in function, and reconstructs any arcs whose attributes have changed. Each arc is drawn with a distinctive pattern, width, and color code to indicate the data type, array dimension, and numeric representation.

Built-in Library

National Instruments' LabVIEW version 2 comes with a large library of VIs (modules) that can take care of most low-level programming details, which frees the programmer to concentrate more on customizing the program. Numerous driver programs for common instruments are also included, and many are available from a growing number of third-party vendors and user-groups. For some unique instruments, the user may have to write his own instrument drivers.

Some of the icons/connectors are shown in Fig. 1.9. As LabVIEW is a programming environment, familiarity is acquired by practice and experience. The user should become familiar with the training material supplied by National Instruments as part of the purchase of the LabVIEW program.

Control Structures

LabVIEW provides five box-like diagram structures and one file-linking structure as shown in Fig. 1.10. The legend for Fig. 1.10 (numbers in circles) is listed in Table 1.2. The top three are quite similar to the "for loop," the "while loop," and the "case structure" (case selector) used in other programming languages, while the sequencer, the formula node, and the Code Interface node are used as follows: to impose the order of execution, evaluate text-like expressions, and link the program to external subroutines. Structure "A" depicts a "Numeric Iteration." This structure is comparable to a "For Loop" in FORTRAN. It executes a specified number of times . The "Repeat Until" structure, or "Conditional Loop," is displayed in B. This loop continues execution as long as the specified Boolean condition is true. Since it checks for the true or false condition at the end of each cycle, it always executes at least once. The "Case Selector," shown in structure C, may contain one or more sub diagrams, also called "cases." One of these "cases" is selected during execution as specified by the input value (which may be either Boolean or numeric scalar). Structure D shows a "Sequencer." This structure holds numerically numbered frames that are executed sequentially. The function of the "Formula Node" in structure E is simply to hold one or more equations. This structure computes the equation sequentially from top to bottom and outputs the result.

Structure F, the Code Interface Nodes (CIN), is equivalent to calling an executable subroutine that is accessible to the program yet is outside the program body itself. The

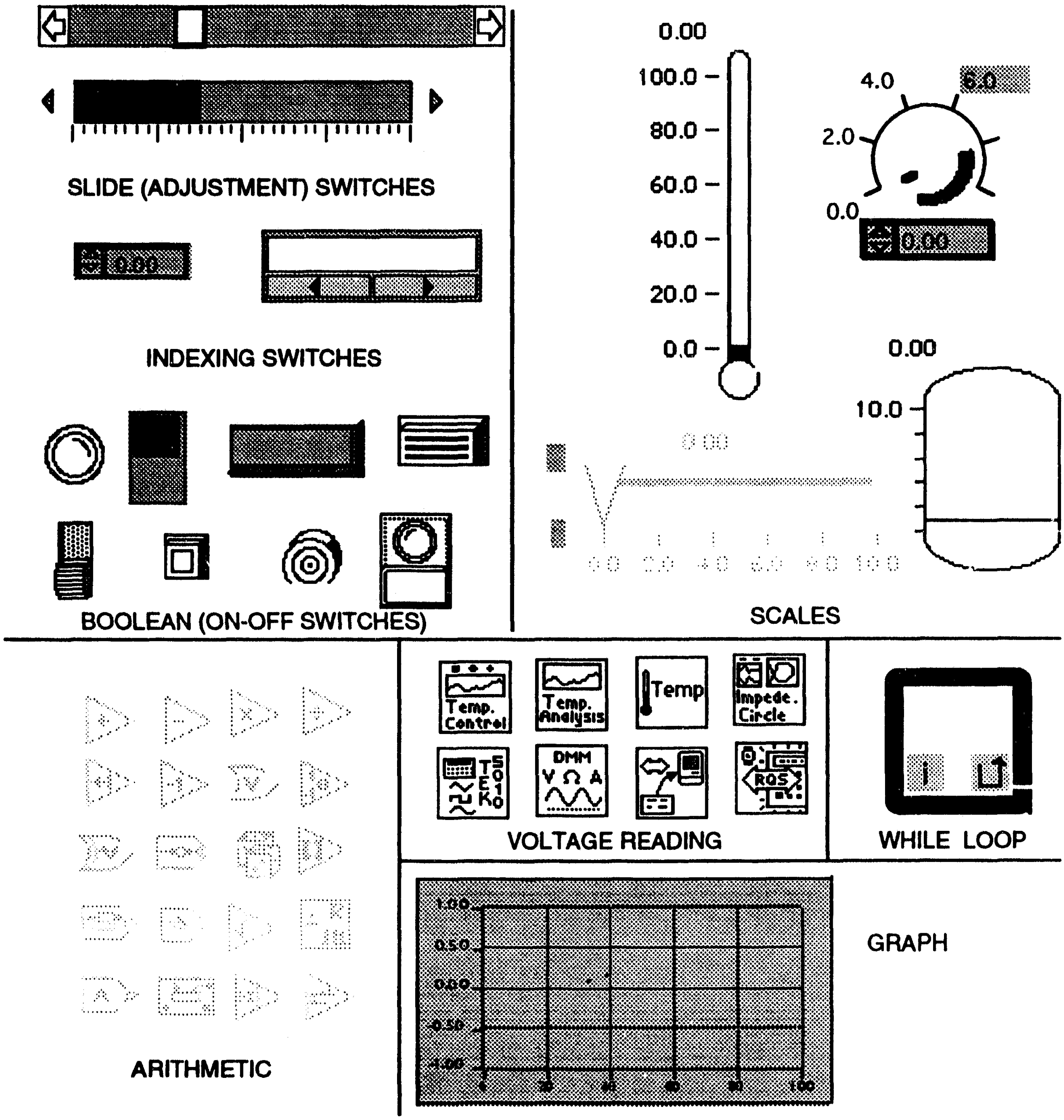


FIGURE 1.9 - Example of icons from LabVIEW 2 Library.

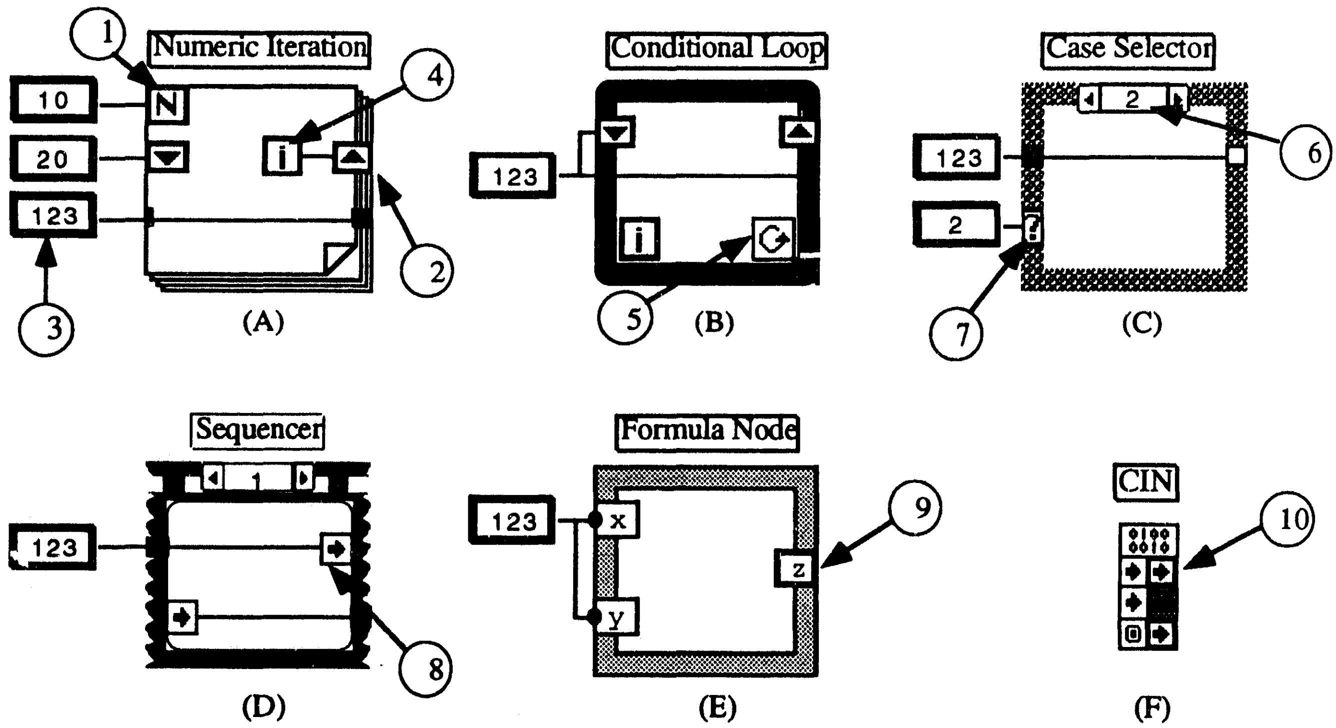


FIGURE 1.10 - Basic control structures in LabVIEW (see Table 1.2 for legend).

TABLE 1.2
LEGEND FOR FIGURE 1.10

1. The N holds the count value (supplied by the constant wired to it) of how many times the loop is to be executed.
2. The arrows on the vertical edges are shift registers, which are used to pass values from one iteration cycle to the next. Any data stored in the up-arrow (right side) is available at the beginning of next iteration cycle through down-arrows (left side). The down-arrow may be initialized by wiring values to be used in the first iteration cycle.
3. This is an example of parameter passing through the structure. A numeric constant, the value 123, is being passed to be used in the structures.
4. The i is the iteration count symbol. It holds the current number of completed cycles.
5. This curved arrow is the loop terminator. It receives the Boolean value of the test condition at the end of each iteration. If the test value is false, it finishes the iteration.
6. This window shows the case being displayed. The right and left arrows allow user to observe different sub diagrams, or cases. The cases may be numeric or Boolean.
7. The question mark receives the case selection information; i.e., the case to be executed.
8. The arrows inside the frame hold the local variables. These variables are used to pass data from one frame to the subsequent frame. The inward arrow indicates a local variable which is wired to receive the value, whereas the outward arrow indicates a local variable which already has a value which can be distributed.
9. An example of the parameter passing method for the formula node. Parameters may be passed through the vertical edges. In this example, z is the output variable and x and y are input variables.
10. The "Code Interface Node," CIN, allows an external routine written in "C" or "Pascal" to be executed here. The parameter can be passed in or out through inward or outward arrows.

subroutine is imported and evaluated using the input parameters provided through the input terminals on the node (arrows in the left boxes). The result is then copied to the output terminals (arrows in the right boxes) where it is available for other program elements. The CIN allows the user to be able to use C, Pascal, or an assembly code language, while still enabling him or her to benefit from object-oriented programming; but most importantly, it allows more efficient dynamic memory allocation for arrays and strings, and this minimizes memory fragmentation.

On-line Help

Information about any of the Sub-VIs can be conveniently obtained by pointing to them with a wiring tool and then pressing certain key-strokes (see LabVIEW software guides). The type of information available on-line includes the name of a module (VI or operator), a brief description of its intended use, and a brief visual/textual description of input/output data and terminal locations. For more details, both the front panel (input/output displays) and the block diagram (program) can be viewed by double-clicking on a module. The entire hierarchy (or a part of it) of a module can also be conveniently viewed.

BIBLIOGRAPHY

Dijkstra, E. W., 1972. Notes on Structured Programming in O. J. Dahl, E. W. Dijkstra, and C. A. R. Hoare *Structured Programming*. Academic Press, London, pp. 1-82.

Kirkman, I. W. and P. A. Buksh, 1992. *Data Acquisition and Control Using National Instruments' "LabVIEW" Software.* Rev. Sci. Instrum., v. 63, No. 1, January, pp. 869-872.

Kodosky, J., J. MacCrisken and G. Rymar, 1991. *Visual Programming Using Structured Dataflow*. Proceedings of the 1991 IEEE (Institute of Electrical and Electronics Engineers) Workshop on Visual Languages, Kobe, Japan, Oct. 8-11. Reprinted by IEEE Computer Society, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1264.

Liles, K., 1991. *Data Acquisition Software Automates Automotive Fuel Injector Test Facility.* Engineering & Management, October/November, pp. 18-21.

Liles, K., 1992. *Data Acquisition—A Mac-Based System for Fuel and Lubricant Testing.* Scientific Computing & Automation, January, pp. 19-23.

National Instruments Corp., 1991a. *LabVIEW 2—Getting Started Manual*, Part No. 320246-01, Austin, TX, April.

National Instruments Corp., 1991b. *Training In-Depth Course on LabVIEW 2,* Version 1.4, Part No. 776393-01, Austin, TX, April.

National Instruments Corp., 1991c. *LabVIEW 2—User Manual*, Part No. 320244-01, Austin, TX, September.

Sethi, R., 1989. *Programming Languages: Concepts and Constructs.* AT&T Bell Laboratories, Murray Hill, NJ, Addison-Wesley, Reading, MA.

Stroustrup, B., 1986. *The C++ Programming Language*. Addison-Wesley, Reading, MA.

Chapter 2

GENERAL DESCRIPTION OF NIPER's AUTOMATION PROGRAMS

INTRODUCTION

NIPER's Lab WARDEN automation software consists of three main interactive sections to provide a central data management system. These sections are: "NIPER MAIN FACILITY, NIPER DISPLAY FACILITY and NIPER GRAPHIC FACILITY." All three of these facilities are integrated such that while the user interacts with one, the others continue to process in the background. Facilities can be easily alternated using a front panel by simply pointing and clicking at the selected facilities. For example, the "NIPER MAIN FACILITY" panel may be activated during the actual scanning of data from pumps, pressure transducers, flow meters, etc., to display the acquired data and report any errors. Then the "NIPER GRAPHIC FACILITY" can be used to display the data graphically. "NIPER DISPLAY FACILITY" panel may be activated to give a visual picture of the process. The following is a brief description of the software. The major functional units of the program are shown in Fig. 2.1, and Fig. 2.2 shows the hierarchical structure and font control panel of NIPER Lab WARDEN, and its components in relationship to National Instruments LabVIEW®.

NIPER MAIN FACILITY

This facility has several features: (1) to read and control instruments; (2) to monitor instruments and when appropriate, issue system errors; (3) to allow reconfiguration of system set-ups, e.g., connect or disconnect instruments, define or redefine allowable ranges outside of which a warning is issued and the system is shut down, establish emergency shutdown sequence, define corrective actions when instrument values fall in the warning range, and set constraints so that instruments are not operated illogically. All of these features are controlled through the "NIPER MAIN FACILITY" panel shown in Fig. 2.3. By pointing and clicking on the buttons in the title bar (see Fig. 2.3), the user may select certain features. Some of these buttons open a new window, allowing new selections. Since these sub-windows have lower priority, they do not interfere with other activities, i.e., data acquisition and control functions continue in the background.

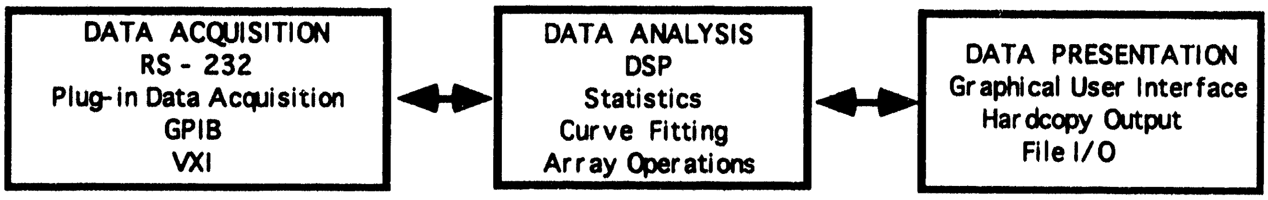


FIGURE 2.1 - Major functional units of NIPER's automation program.

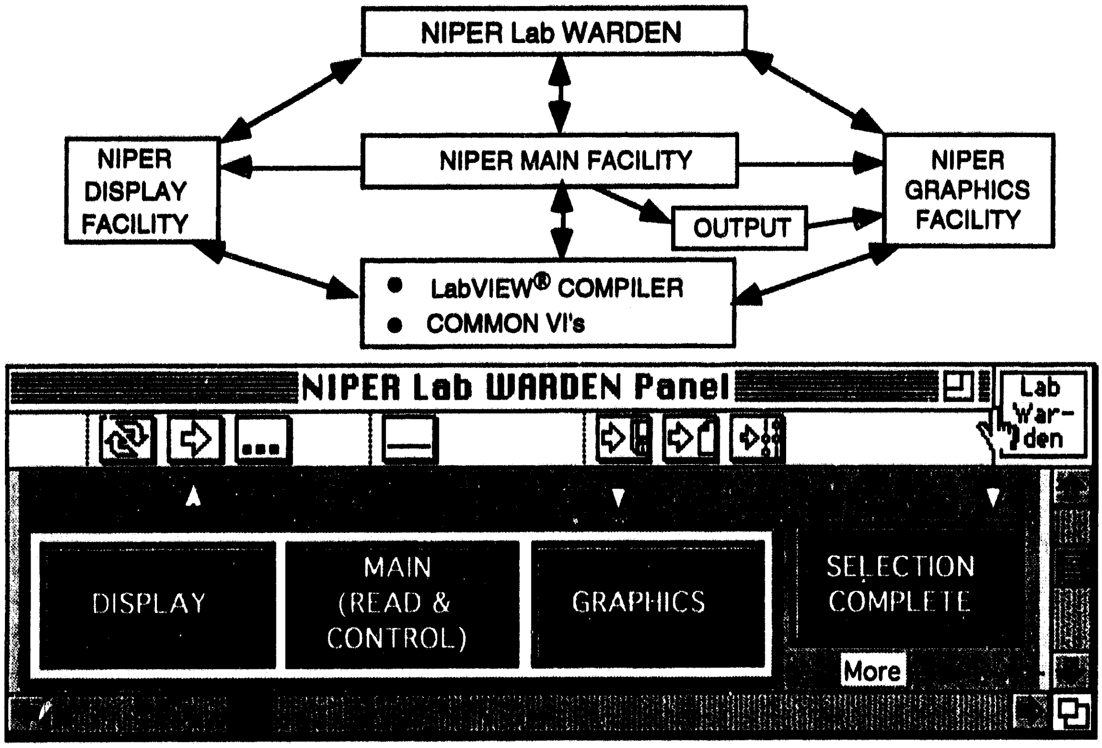


FIGURE 2.2 - Hierarchical structure and front panel of NIPER Lab WARDEN.

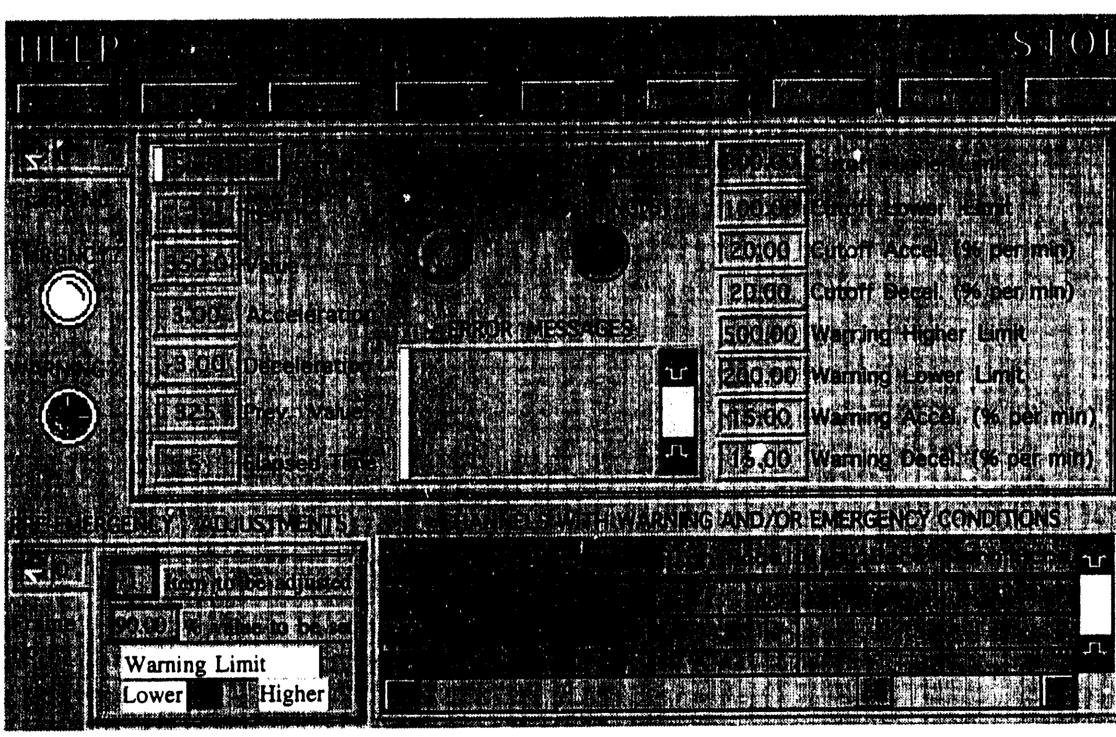


FIGURE 2.3 - NIPER MAIN FACILITY panel.

NIPER DISPLAY FACILITY

The front panel for this facility is shown in Fig. 2.4. "NIPER DISPLAY FACILITY" allows the user to pictorially view the status of remote-sensing instruments such as pumps, controllers, balances, temperature sensors, pressure sensors, fluid flow controllers, and alarms through the computer panel. This facility can be used, for example, to set built-in logic to check the validity of instrument control reset commands and to disallow them when unsound, e.g., it can be set not to allow a pump to run if the outflow valve is closed, thus reducing the chances of avoidable accidents and failures. Similarly, the facility can be configured to continuously monitor the test progress by material balance, i.e., the injected fluids and discharged fluids are continuously weighed and compared. If a discrepancy beyond the set acceptable limits is found, the facility warns the user of possible leaks or other mishaps in the system. When a situation falls outside the defined ranges, and no responsive action takes place, this safety feature can also be used to automatically turn the system off.

"NIPER DISPLAY FACILITY" is an optional facility that does not need to be running during automation process (reading/controlling/monitoring instruments). Its main usefulness is

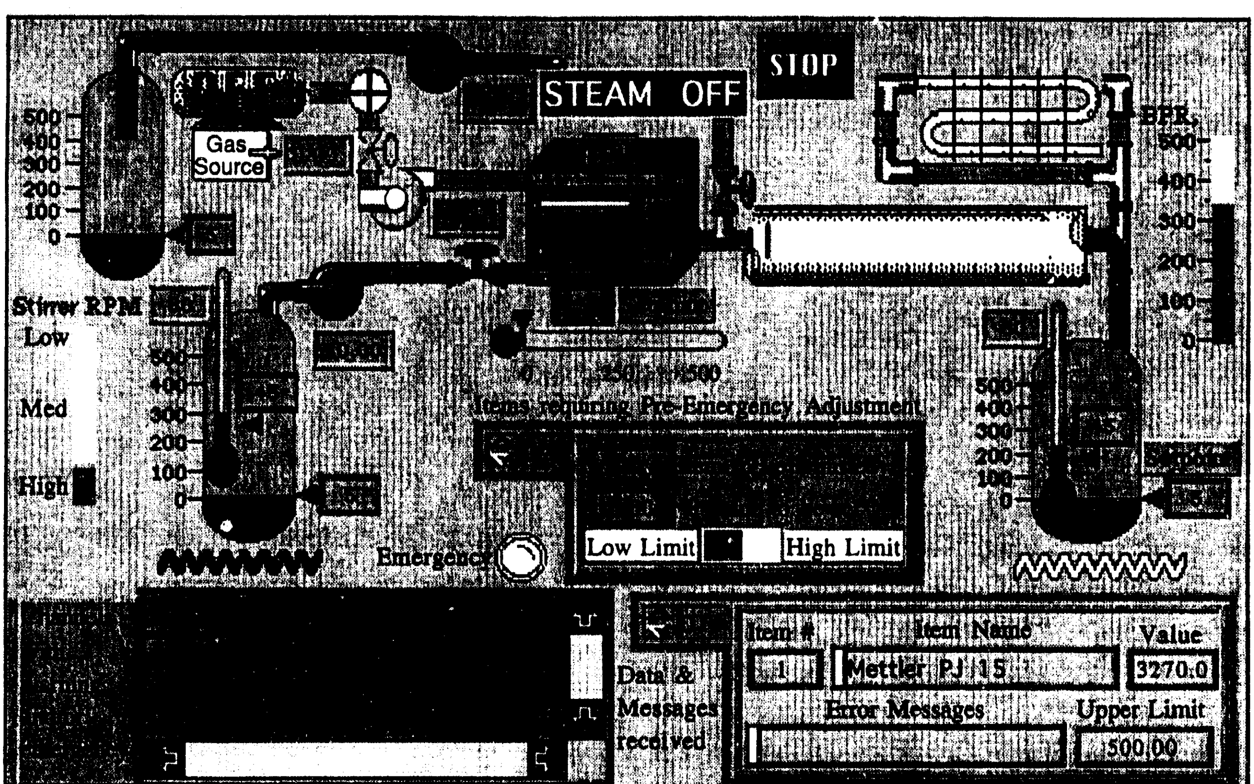


FIGURE 2.4.- NIPER DISPLAY FACILITY panel.

in quick recognition of possible trouble-spots that need immediate attention since a visual examination of the status of items is faster than extracting it from a cluster of data. Its use is recommended when enough RAM memory is available and when the operator is only partially attentive because of other activities.

NIPER GRAPHIC FACILITY

The front panel for the NIPER GRAPHIC FACILITY is shown in Fig. 2.5. This facility allows the user to select or manipulate any portion or combination of data obtained from a current or previous experimental run, e.g., analyze, compare, plot, curve-fit, add, delete, and copy interactively. The selected data can be printed or exported to other programs such as Microsoft Excel. Data can also be sent to a full-size graph where they can be edited using a mouse, i.e., a point or a segment of curves can be deleted, relocated, added, or copied/pasted from another graph. This interaction with data for graphical display or printing can be conducted independently, or during runtime while the program is collecting data and monitoring the process (laboratory or the pilot plant) in operation. The 3-D graphic display shown in Fig. 2.6 is an

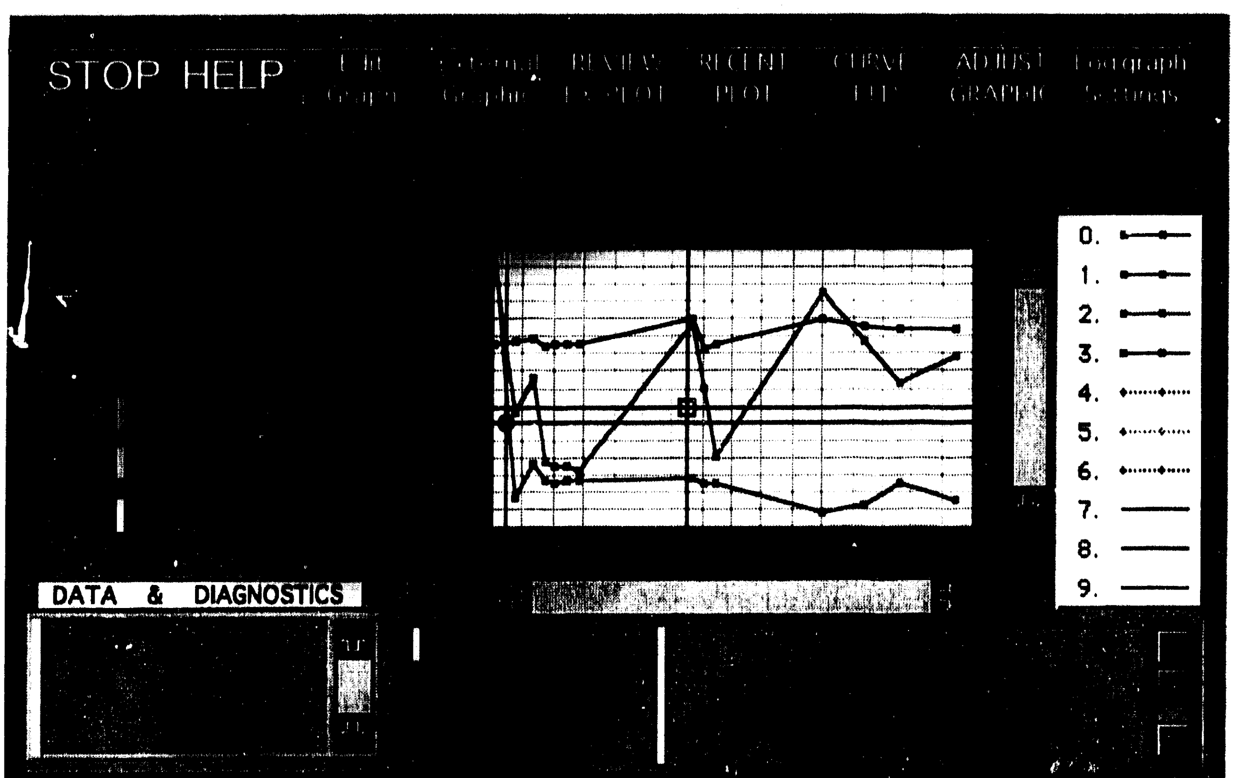


FIGURE 2.5 - Front panel of NIPER GRAPHIC FACILITY.

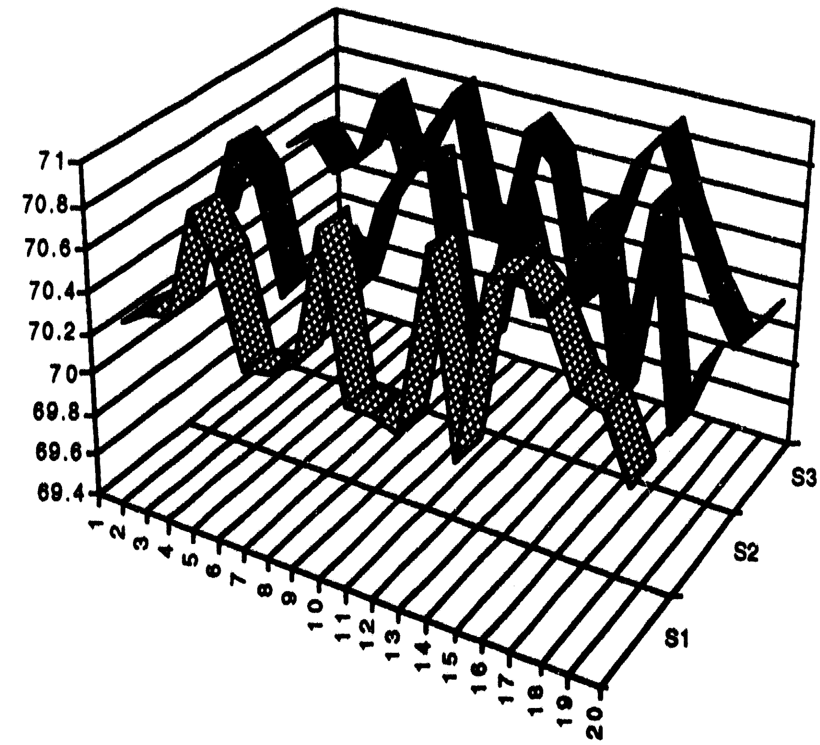


FIGURE 2.6 -A sample snapshot of automatic 3-D display of data in Microsoft Excel. DeltaGraph Professional and Spyglass Transform are similar linkable presentation graphics programs for display.

example of the data export and communication capability with other applications. In this example, the data were reviewed in "NIPER GRAPHIC FACILITY" and then exported by pressing the "External Graphic" button, which opened a Microsoft Excel file containing the new data. The data were plotted, and the chart was automatically rotated with different attributes such as different angles, aspect ratios, etc. Then, the chart and the Microsoft Excel application were closed, and the control was transferred to LabVIEW. Lab WARDEN continues to operate in the background acquiring (or waiting for) new data and checking for error and safety messages.

Excel is one of many applications that can be integrated with LabVIEW. In fact, any application that supports Microsoft System 7's feature of Dynamic Data Exchange (DDE) can be integrated, and can link, subscribe, and publish to or from other files. Also, any application that allows macro-expansion (sometimes called scripting) is a good candidate for integration. Some of the commercial applications that have these capabilities are Excel™, DeltaGraph Professional™, and Spyglass Transform™. Spyglass Transform is an exceptional candidate for enhancing LabVIEW's graphic capabilities since it can display diffused-color surreal graphics by filling in interpolated data points along with the actual data in a multidimensional spatial field.

"NIPER GRAPHIC FACILITY" is also an optional facility, i.e., it does not need to be running during automation (reading/controlling/monitoring instruments). Its sole purpose is to provide off- or on-line (real time) data selection capabilities and advanced graphics for review and printing purposes. The use of this facility should be minimized while a test is in progress, unless enough RAM memory is available.

The software that is the focus of this report is only a small component of the overall automation process of a laboratory or pilot plant. Often, the hardware interfacing with the computer is quite elaborate. The operational details of control instruments, pumps, valves etc. (the physical units of the system) need to be understood before a proper connection with the electronic boards can be made. The plug-in boards, as shown in Fig. 2.7, are the bridges between instruments and the computer. Therefore, understanding and proper installation of these boards is required. Some instruments may have unique operational logic and require simple programming in LabVIEW to interface with NIPER's programs.

INSTALLATION

Installation of LabVIEW

Installation of National Instruments LabVIEW requires at least 16 megabytes of hard disk space and between 20 and 30 minutes for installation. To begin installation, insert LabVIEW disk 1 and double-click on the LabVIEW installer icon. A folder called LabVIEW 2 is created and will contain all of the files. Change this name if you wish to install the software in a folder with a different name. Press the install button to install LabVIEW completely, or hit the custom install to install selected items. To run LabVIEW refer to National Instruments "Getting Started Manual" (1991a) tutorial. Only the LabVIEW Application, Control Library, and the Getting Started, Analysis and Utilities portions of the VI library are needed. Insert the disks as they are requested by the prompts within the installation procedure.

Installation of NIPER Lab WARDEN

Installation of NIPER Lab WARDEN requires at least 5.6 megabytes of hard disk space and 3 megabytes of RAM. Additional hard disk space and RAM are strongly suggested and are required when interfacing with other commercial software products. The example problems in chapter 9 were solved using National Instruments boards configured with those of other manufacturers (hardware). Boards from other manufacturers may have different specifications that must be addressed before using NIPER Lab WARDEN. Before ordering any board, its compatibility with LabVIEW should be ascertained, as well as the availability of a suitable slot and power capacity in the current computer must be determined.

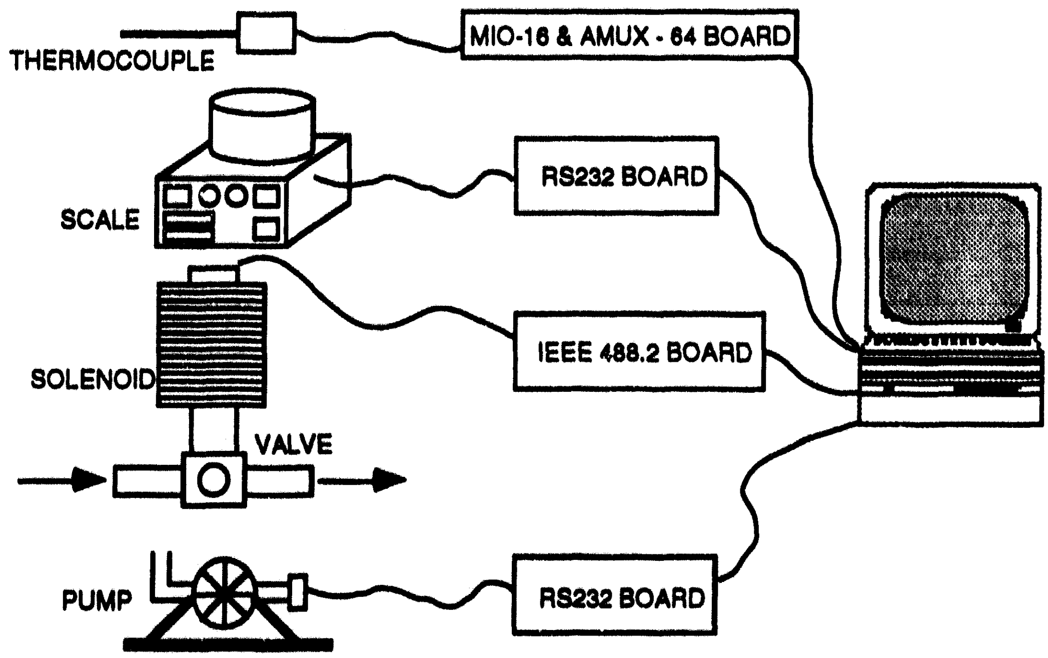


FIGURE 2.7 - The software developed is only part of the automation of equipment. The boards are the bridges between equipment and computer.

The following steps will install NIPER Lab WARDEN properly: Consult the teach text files on the program disks for the latest updates for installation or changes in the program.

1. Make sure you have the two disks "NIPER Lab WARDEN Vol. 1 of 2" and "NIPER Lab WARDEN Vol. 2 of 2" that accompanied the NIPER AUTOMATION MANUAL (this manual). You need to have the National Instruments "LabVIEW" version 2.2 or higher to run these programs. You need to have 7.6 MEG for the NIPER Lab WARDEN + memory for LabVIEW (Minimum 2.5 MEG) to run. You may obtain a run-only version that needs only 5.6 MEG (including LabVIEW), or you can convert the programs in these disks to a run-only version. See the National Instruments manual if you need run-only version.
2. Make a copy of each disk on the two floppy disks or on hard disk and work from them. Keep the original disks locked in a safe place.
3. Create a new folder on the desktop or on the internal hard drive. Name the folder "NIPER Lab WARDEN." (You may name the folder a different name but you will have to rename the folder pathnames in three separate locations as the programs run. It can be easily done.)
4. Insert either disk into your computer.
5. Click open the disk that was inserted if it is not already open.

6. Click open the icon in the active window "NIPER Lab WARDEN Vol. 1(or 2) of 2." It will open another window labeled "Select Destination Folder...". This window allows you to select the folder in which you would like to store the NIPER Lab Warden program from the inserted disk.
7. Select the folder "NIPER Lab WARDEN" (or whatever you have named it) that you just created in step 3 above. To the right of the window, there is a button "OPEN". Click on this button to open the folder "NIPER Lab WARDEN". Alternately, you can double-click on "NIPER Lab WARDEN" to open it.
8. If the window title bar is showing "NIPER Lab WARDEN" (or whatever you have named it), click on the "Extract" button on the right border of the active window.
9. Wait for the "Extraction" procedure to be completed.
10. If the disk contains more than one compressed file (names ending with .sea), follow steps 3-9 for each file to install the contents on "NIPER Lab WARDEN" folder. It is best to have all the files in the same folder.
11. After installing the contents of the first disk, insert second disk and follow steps 3-9 to install the contents of the second disk. Again, it is best to put these contents in the same folder that you put the contents of the first disk.
12. (Optional) If you have system 7 or higher, you can access your program faster by listing "NIPER Lab WARDEN" in the apple menu. Find the file "NIPER Lab WARDEN," make an alias of this file, and drag the alias to the "Apple Menu Items" folder. You may use ⌘+F to find the file, select "Make Alias" from the pull-down "File" menu to make alias of the file highlighted (selected), and drag the file by pointing, click-holding, and moving the file to its intended destination.

The installation in now complete. However, take these additional steps to make sure it has been properly installed.

a. Re-start the computer.
b. Click on "NIPER Lab WARDEN" icon, or select it from the apple menu (if you have completed step 12 above). The file should open and the "Finder" icon on the top-left corner of the apple menu bar should change to the "LabVIEW" icon, and the menu bar titles would also change.

If it has not done so, or if a message something like "The document NIPER Lab WARDEN" could not be opened, because the application program that created it could not be found" appears, then open the LabVIEW application itself and open the "NIPER Lab WARDEN" within the LabVIEW. You may find files using command described in Step 12 of these instructions. If you can open it successfully, then the installation is proper. If not, remove and reinstall "NIPER Lab WARDEN" again from scratch.

c. Repeat step b of these instructions with "NIPER MAIN FACILITY, NIPER DISPLAY FACILITY, and NIPER GRAPHIC FACILITY."

BIBLIOGRAPHY

National Instruments Corp., 1991. *LabVIEW 2—Getting Started Manual*, Part No. 320246-01, Austin, TX, April.

Chapter 3

DESCRIPTION OF NIPER MAIN FACILITY PANEL

A QUICK TOUR OF THE NIPER DATA ACQUISITION FACILITY

"NIPER MAIN FACILITY" is the core of the software program that enables the user to interface with the instruments (read data, send control commands define the sequence of instrument as reset or set desired operating status), analyze raw data received from the instruments, compare the data with the user-defined ranges, and display the status of each instrument (warning, error, and conditions). It also provides two safety features: it can correct a warning situation by adjusting the value of certain instruments according to the user specifications, and it can systematically shut-down the experiment by following the prescribed sequence. A description of the "NIPER MAIN FACILITY" panel (Fig. 3.1) and its sub-panels is given below. Circled reference numbers refer to the description of various attributes described in Table 3.1.

When "NIPER MAIN FACILITY" is operated, a new window named "NIPER Indicator" is opened whose front panel is shown in Fig 3.2. The circled reference numbers refer description to the various elements presented in Table 3.2. Clicking on the "Accept Changes?" or "Reject Changes?" opens the "NIPER Control" panel (Fig. 3.3). Finally, clicking on "Accept Changes" or "Reject Changes" closes this window and re-displays "NIPER MAIN FACILITY" panel.

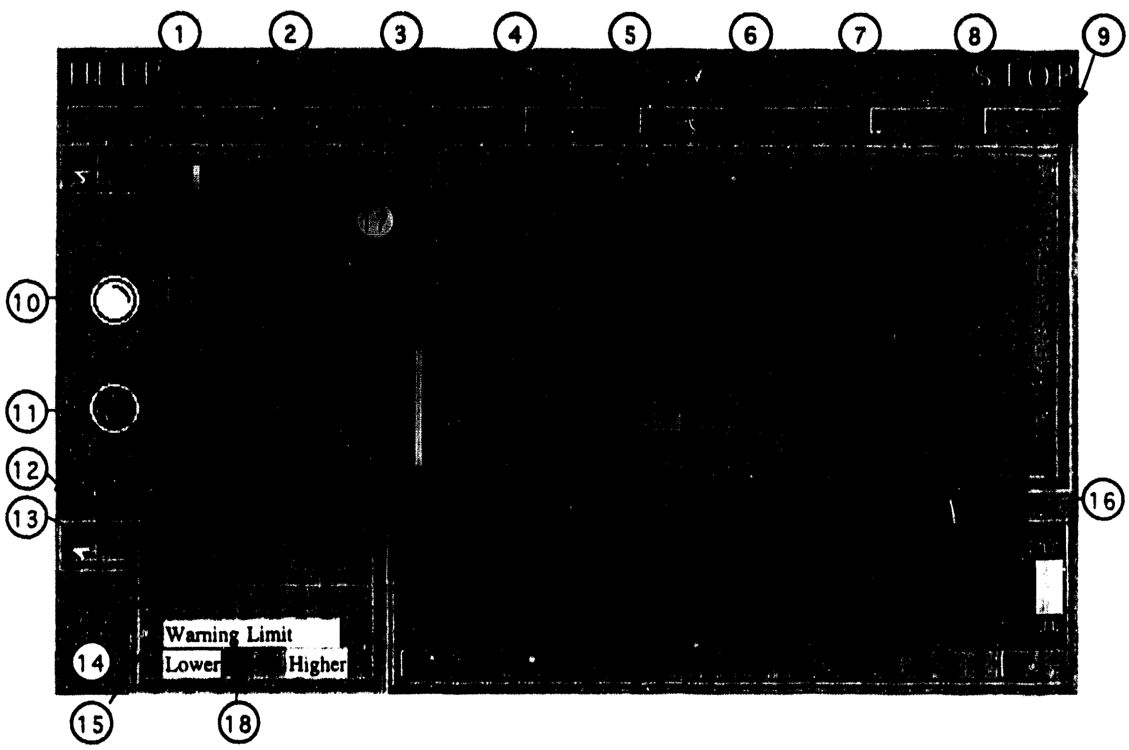


FIGURE. 3.1 - Front panel of "NIPER MAIN FACILITY."

TABLE 3.1

LEGEND FOR FIGURE 3.1

1. This button opens the NIPER HELP panel which provides information about LabVIEW and object-oriented programming.
2. The ADJUST CONTROL button allows the user to adjust the parameters pertaining to various instruments and indicators of the experiment. Pressing this button first opens the "Indicator" panel. On this panel, the user is able to adjust the various controls of the experiment and select various instruments. After the desired changes have been made and the "Reject Changes" and/or "Accept Changes" button has been pressed, the "Control" panel appears. Here changes to the hardware configuration can be made.
3. The LOG CONTROL button allows the user to run the test using a previously stored setting from the log library. This button can also be used to save one or more settings in the log library for future use. These functions are useful when the program is being used with more than one set-up and the user wants to switch from one to another without redefining default values.
4. When the program encounters I/O errors so frequently that the normal execution of the program becomes impossible, a dialogue box appears allowing the user to continue by disallowing the display of I/O messages. Pressing this button once will allow the program to revert to the original status of dialog display.
5. By selecting the GET NEW DATA button, an instantaneous reading of the instruments is taken. This reading is supplemental to the readings taken at due scan intervals, adding one extra set of readings to the data. This button is useful when the user wants to know the current status of the run.
6. The CHANGE DIRECTORY button allows the user to specify the directory where various data, error, and log files are to be placed or searched. This allows the user to organize data in different directories for more efficient retrieval.
7. The DATA UPDATE button allows raw data to be exported to a global VI. Pressing this button is necessary if "NIPER Lab WARDEN" is being run and needs to receive data. If data export is not required, it is recommended that this button not be pressed to avoid efficiency loss.
8. The GRAPH UPDATE button allows graphic data to be exported to a global VI. Pressing this button is necessary if "NIPER GRAPHIC FACILITY" is being run so that it can receive data. If data export is not required, it is recommended that this button should not be pressed to increase efficiency.
9. The PRESS TO STOP button stops the operation being run. All of the collected data is stored, and the total number of ex-runs is incremented by one. Stop does not quit NIPER Lab WARDEN or LabVIEW. To exit NIPthese, the user must to quit these programs. Stop also resets the control instruments to their defined termination values or status in a systematic sequence.
10. This emergency light flashes when the data readings do not lie within the specified acceptable range of parameters (initially set by the user). If action is not taken when the emergency light is flashing, the automation program will automatically shut down the entire system in an orderly fashion.
11. The warning light is less serious than the emergency light. This light flashes to warn the user that data readings are in the warning range (initially set by the user). A set of actions is taken by the computer to rectify the situation. These corrections involve resetting some control-instruments.
12. This box contains a list of instruments being automatically reset by the computer due to a warning condition, if any.
13. The frame numbers are the order in which the instruments (in box #12) will be reset.
14. This number identifies the instrument (in box #12) to be reset. The number corresponds to the frame number of the "NIPER Control" front panel box on which the item is described.
15. This number determines by what percentage the current values will be reset. or implies the instrument will be turned off, and 999/. implies it will be turned on.
16. This box contains information about the instruments experiencing warning or emergency conditions, if any.
17. The Summary for Diagnostics box contains information about each indicator-instrument. It is useful in warning and emergency situations when a quick diagnostic of the situation is desired to decide whether to override the corrective actions soon to be taken by the program.
18. This display Boolean switch shows whether the instrument displayed in box #14 will be reset to the value shown in box #15 as a result of lower limit warning conditions or higher limit warning conditions.

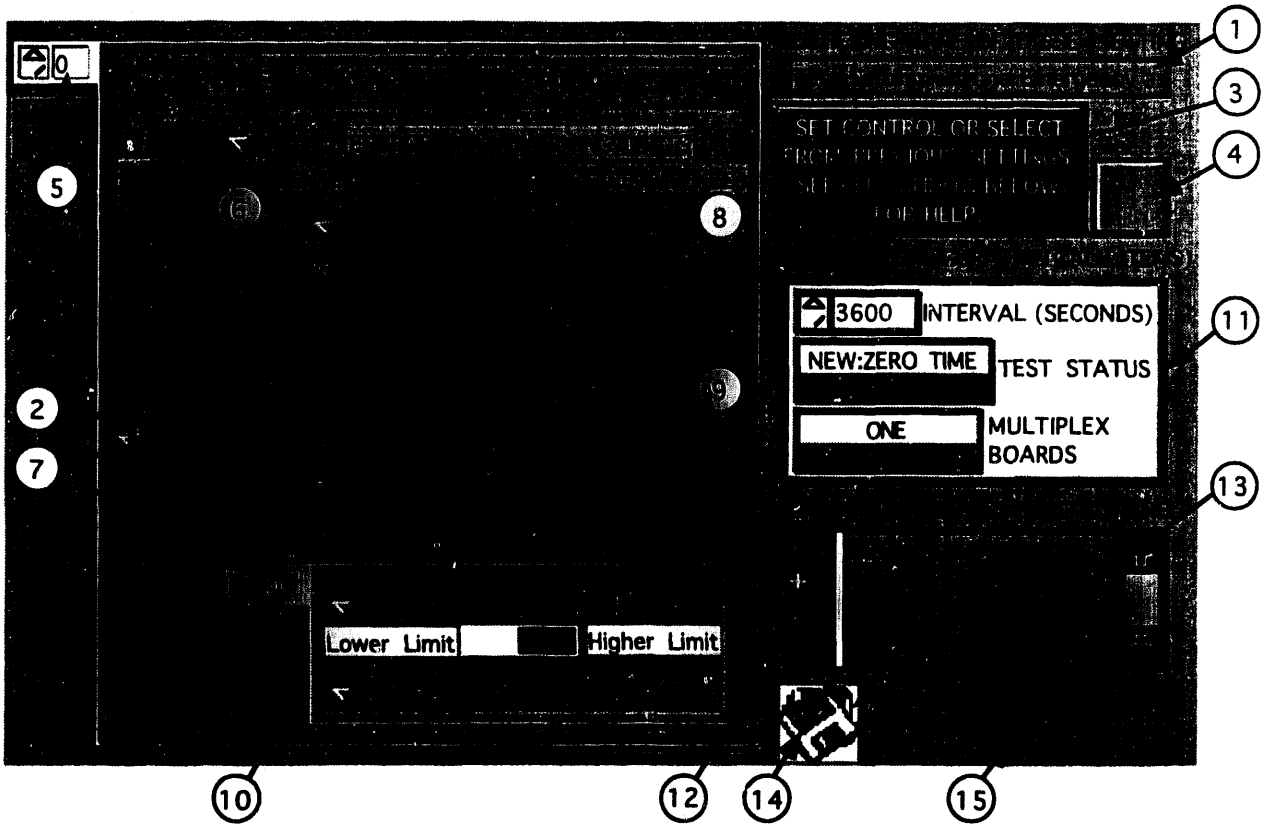


FIGURE. 3.2 - Front panel of "NIPER Indicator" VI.

TABLE 3.2
LEGEND FOR FIGURE 3.2

---

1. This box contains the name of the experimental set-up. This name is arbitrary and does not alter the functionality of the program of the displayed instrument in the list.
2. This button stops all active VI's. It can be used to halt the experiment at this stage.
3. This button serves as an audio-visual reminder to the user that an input is needed here to proceed.
4. This button stops the audio-visual reminder explained in number 3 above.
5. The frame number corresponds to the sequence number of the displayed instrument in the list.
6. This is the item number that the user assigns to his instrument whose name appears in the ITEM NAME box This is an arbitrary number which does not alter the functionality of the program. However, -1 or any other negative integer will deactivate it from the list.
7. This display contains information about how to communicate with the instrument, i.e., board number, channel number, and driver VI.
8. The emergency cut-off parameters are set in this box.
9. The warning parameters are set in this box.
10. Here the user can set the value of each instrument to be reset by a certain percentage in a warning condition. The resetting is done in the order of the frame number.
11. This display allows the user to set various controls for the experiment. The TEST STATUS sets how the initial data is to be read (data can be read at time zero, a continued time, or appended to a previous run). The number of multiplex boards being used can be set here. Also, the time interval at which each data reading is to be taken by the instruments is set here.
12. The "+" and "-" buttons are used to select the information to be displayed.
13. This provides helpful information about the use of the indicator panel.
14. This is pressed to accept the current settings.
15. This is pressed to reject any changes that may have been made.

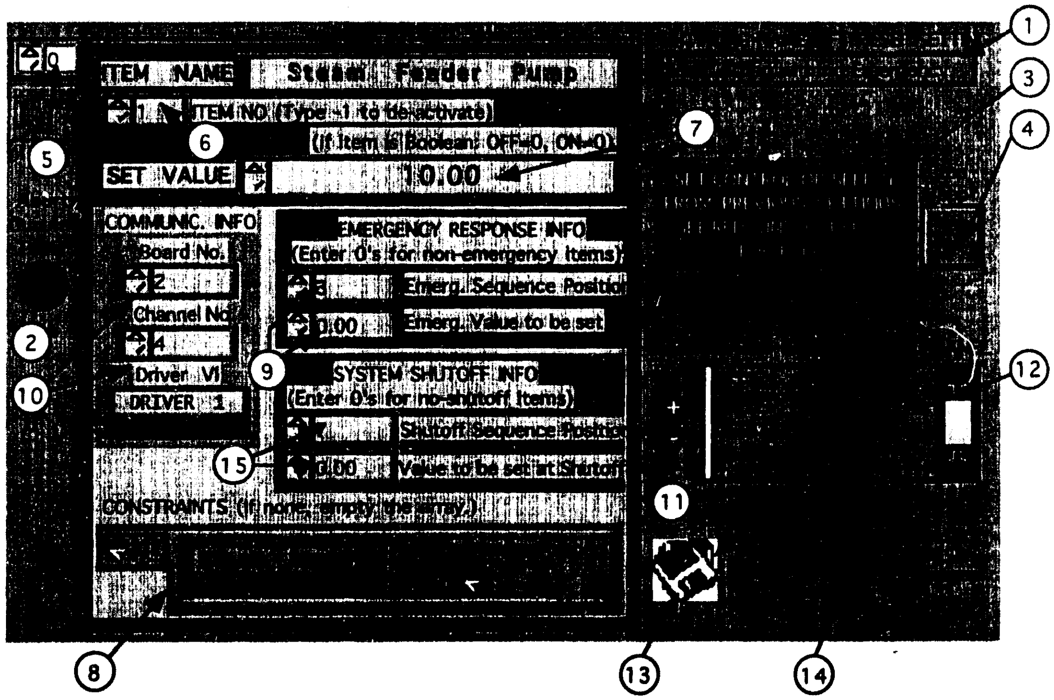


FIGURE 3.3 - Front panel of "NIPER Control" VI.

TABLE 3.3
LEGEND FOR FIGURE 3.3

1. This box contains the name of the experimental set-up. This name is arbitrary and does not alter the functionality of the program.
2. This button stops all active VI's. It can be used to halt the experiment at this stage.
3. This button serves as an audio-visual reminder to the user that an input is needed here to proceed.
4. This button stops the audio-visual reminder explained in number 3 above.
5. The frame number corresponds to the sequence number of the displayed instrument in the list.
6. This is the item number that the user assigns to his instrument whose name appears in the adjacent box This is an arbitrary number which does not alter the functionality of the program.
7. The user can set the default value at which the instrument will operate unless otherwise changed by the program or the operator. For ON/OFF, use 999 to activate items, even though any non-zero number will activate the instrument while zero will deactivate.
8. This box allows the user to set constraints on the instruments. Under a constraint, an instrument's activation is made conditional upon the active/inactive state of other instruments. Constraining is done with two controls inside the box. The item on frame number box corresponds to the frame number (box #5) on which the selected instrument is located.
9. These two boxes enable the user to establish an emergency response where the values of the selected instruments are reset during the emergency. The Emerg. Sequence position indicates the position of the instrument (if included) in the sequence of emergency reset responses. The Emerg. value to be set is the value at which the instrument will reset also if it is included in the emergency response sequence.
10. This display contains information about how to communicate with the instrument, i.e., board number, channel number, and driver VI.
11. The "+" and "-" buttons are used to select the information to be displayed.
12. This provides helpful information about the use of the indicator panel.
13. This is pressed to accept the current settings.
14. This is pressed to reject any changes that may have been made.
15. These two boxes enable the user to establish a shut-off response where the values of the selected instruments are reset when stop button is pressed by the user in NIPER MAIN FACILITY. The shut-off Sequence Position indicates the position of the instrument (if it is included) in the sequence of shut-off reset responses. The value to be set at shut-off is the value at which the instrument will set to if it is included in the shut-off response sequence.

Chapter 4

OPERATION OF NIPER's MAIN FACILITY

INTRODUCTION

Features

This software (facility) allows the user to:

(1) Configure the software for the desired experimental set-up,

(2) Save several experimental configurations for future recall,

(3) View and recall a previously saved configuration for use in the current run,

(4) Scan and view data from the selected instruments at fixed intervals or as desired,

(5) Control (change settings) selected instruments at start, and set stop, warning, emergency conditions as desired,

(6) View complete status of the test including warning and error messages,

(7) Provide data and run status information to other NIPER facilities and commercial programs.

"NIPER MAIN FACILITY" is a self-sufficient facility that does not require any other NIPER facility to accomplish its features. When run, it asks the user to provide information about the instruments and the design (control) parameters of the experiment. Such information can be saved and retrieved freely, avoiding repetition. This allows multiple experimental set-ups to be handled easily by a single program, by saving and recalling the settings for each set-up.

The facility provides complete information about the status of the test. This is achieved by comparing the recently acquired data of each instrument with respect to its defined limits and the instrument's previous value. Besides providing audio-visual warnings and error messages to the operator screen, such messages can be sent to remote terminals. The facility can be configured to take corrective actions when the value of an instrument exceeds user preset limits (warning limits). If a critical condition exists as defined by the user in their configuration, the facility may automatically initiate a user-defined systematic shut-down procedure. The facility also allows the user to define a start-up sequence, i.e., the reset value (or ON/OFF status) and its position in the sequence can be defined for as many instruments as desired. Similarly, the user may also define a shut-down sequence to be initiated at the termination of the test.

Technical Information for Programmers

The program executes in three stages: (1) initialization, when files are opened and are initialized to the default parameters, and the control-instruments are reset to their start-up values sequentially; (2) a loop, in which the program handles service requests such as opening the front panels for user interaction; and (3) termination, where the system is shut-down systematically, i.e., the control-instruments are reset to their stop values sequentially, files are closed and the current status is saved. This stage begins when the "STOP" button is pressed. At the beginning of each loop, the program notes the lapse time after the last reading. If this lapse time has exceeded the user-defined scan interval, or if the "GET NEW DATA" button on the front panel is pressed, data from each instrument are read and saved in one of the open files. These data are then analyzed and compared to the defined ranges. The analysis of data and emergency/warning conditions are reported on the front panel. If an emergency condition exists on any item, all control-instruments are reset to their emergency status as defined in the "NIPER CONTROL PANEL" VI. If a warning condition exists on one or more items, the corrective actions are taken according to their preset sequence defined in the "NIPER INDICATOR PANEL" VI. These corrective actions involve resetting the control-instruments to an adjusted value.

Since the data are stored in files as soon as they arrive, a copy is immediately available for user handling. The user can perform any operation on the data using any software that handles "tab-delimited-text" format, e.g., Microsoft Word and Excel. Manipulating these data does not interfere with LabVIEW operations such as data acquisition and instrument control. The original file containing the data is locked and not available for editing while the program is running.

OPERATION

Displays and Controls

Control Bar

The top control bar in the "NIPER MAIN FACILITY" contains 9 buttons with the labels and functionality described below:

(1) "HELP" This button opens a panel (Fig. 3.3) which describes general information about LabVIEW and this program. Itemized help information is also available which is described later.

(2) "ADJUST CONTROLS" This button opens two panels (Figs. 3.2 and 3.3) which allow the user to feed information about the desired communication with instruments. The first panel is for indicator-instruments (sensors/gauges) and contains information about the instrument such as identification (number and name) and the set values (high/low and accel/decel limits). It also contains information about corrective actions to be taken in a warning condition. A corrective action is defined as adjusting the current value of a control-instrument by the specified percentage, or changing its current status (ON/OFF). It also contains general information pertaining to all instruments, i.e., test status, number of multiplex board in the computer, and scan interval.

The second panel is for control-instruments. It contains information about the instrument identification (number and name), the communication port (board number is the slot number in which the board is physically plugged into the computer), and it also contains information about the instrument's set values at start-up, at shut-down, in an emergency, and the sequence for emergency and other response actions. It can also be used to set constraints, e.g., to disable turning an instrument ON when another instrument is OFF.

(3) "LOG CONTROLS" This button allows selection among a set of instrument settings previously stored by the user. This is useful when the user wants to use the program for multiple sets of experiments. In this situation, each setting is saved with a unique name for quick identification. Any of these saved settings can be quickly retrieved later. This button can also be used to store a new set of instrument settings. The function of this button is to open the front panels of the "NIPER Control" and "NIPER Indicator" VIs without going through the finder to open them. Alternately, these files can be opened directly from the finder. Once these files are opened, the settings can be saved and/or retrieved.

(4) "ENABLE DIALOGUE" This button allows the file I/O error messages to be displayed in dialogue boxes. When I/O messages are displayed, the user is given an option to discontinue displaying such messages to avoid disruption of the test in progress. This button is used to remove this previously set "non-display" flag. This error-recovery mechanism is built-in to allow the user to continue the test even if there are errors in the system.

(5) "GET NEW DATA" Data are scanned once every time this button is pressed. It is used to examine the current status of the test. Pressing this button adds an additional set of data besides those scanned at preset intervals.

(6) "CHANGE DIRECTORY" This button opens a panel containing information about the location of directories. The most likely cause of I/O error messages is an incorrect path description of the directory containing the needed file. This button is used to change the directory paths.

(7) "DATA UPDATE" Pressing this button allows data to be sent to the "NIPER DISPLAY FACILITY" when it is running. If "NIPER DISPLAY FACILITY" facility is not running, pressing this button will have no effect except slightly reducing computer efficiency. Press this button only when "NIPER DISPLAY FACILITY" is running.

(8) "GRAPH UPDATE" Pressing this button allows data to be sent to the "NIPER GRAPHIC FACILITY" when it is running. If "NIPER GRAPHIC FACILITY" is not

running, pressing this button will have no effect except slightly reducing computer efficiency. Press this button only when "NIPER GRAPHIC FACILITY" is running.

(9) "PRESS TO STOP" This button allows the main facility to stop execution after carrying out a normal shut-down procedure (i.e., reset selected control-instruments in prescribed sequence), completing the current cycle and closing all files. Using this button to stop execution is safer than using "Operate" menu, pressing '⌘+.', or closing the file—all of which terminate the program immediately without resetting instruments. It is, therefore, recommended that the execution be stopped using this button to avoid unknown side-effects. When this button is pressed, user is given options (through dialogue boxes) to exit with or without system shut-down. Like emergency shut-down, the normal shut-down procedure (sequence) is defined (or selected) by user through "NIPER CONTROL" front panel during the run.

Other Controls and Displays

In the "NIPER MAIN FACILITY," the front panel contains additional controls and display capabilities including:

"EMERGENCY?" This indicator lights up when one or more scanned data fall outside the allowable range. When this happens, user can quickly browse through "Summary for diagnostic" indicator box to locate the channel(s) having emergency situation(s) and examine the pertinent data to decide whether to override emergency shut-down.

"WARNING?" This indicator lights up when one or more scanned data is in a warning range. When this happens, the program may take some corrective actions by resetting selected control-instruments. A list of these corrective actions is displayed in "Channels requiring pre-emergency adjustments" box.

"Channels requiring pre-emergency adjustments." This indicator box contains a list of the instruments that are being reset to the indicated percentage of their current value. The box remains empty except in a warning situation. Zero percent implies the instrument will be turned-off, and 99.9% implies it will be turned ON.

"Summary for diagnostic." This box contains comprehensive information about each indicator-instrument. It is useful in warning and emergency situations—when a quick diagnostic of the situation is desired—to decide whether to override the corrective actions soon to be taken by the program.

"Channels with Warning and/or Emergency Conditions." This table provides a summary of the status of the test. The name of items with warning or emergency conditions are displayed along with their numbers, error messages, and values in appropriate columns. If no such conditions exist, the table is blank. The scroll windows can be used to browse through the table.

OPERATING INSTRUCTIONS

To Open the Program

Open the file "NIPER MAIN FACILITY" by one of the following instructions: (a) if "NIPER Lab WARDEN" front panel is not already open, open it. Run the VI, press "Read & Control" button, then press "SELECTION COMPLETE" button. It will automatically open "NIPER MAIN FACILITY" VI, (b) find and double-click on the file named "NIPER MAIN FACILITY," or (c) open this file within the LabVIEW program by using "open" from the pull-down menu. The "NIPER Lab WARDEN" VI can be used to conveniently go back and forth between facilities. It can be opened from the pull-down menu under the apple icon.

To Get Information

Run the "NIPER MAIN FACILITY" VI if it is not already running. Press "HELP" button located in the top control bar. A panel will be opened that describes general information about LabVIEW and this program. This help information is not topic-by-topic. If you need more than one minute to read the information, keep pressing the "CONTINUE" button within 60 seconds, the window will automatically close after about 1 minute. When finished, press the "DONE" button to go back to "NIPER MAIN FACILITY."

The information obtained using the "Help" button is general. To get itemized information, point the cursor towards the item, press ⌘ and select "description" from the pop-up menu. A box will appear providing details about the item.

To Make a Run

Open (load) the file "NIPER MAIN FACILITY" (The section "To Open the Program" provides instructions). The program automatically begins execution. If it is not running (i.e., the arrow on menu bar is not blackened), run it by clicking on this arrow. It will open the "NIPER Indicator" panel.

If this is not your first run and you have already set the default values:

(a) Press "Accept Changes?" on the "NIPER Indicator" panel already open.

(b) A second panel "NIPER Control" will open up. Press "Accept Changes?" again. The program will begin to reset the control-instruments and will go back to the "NIPER MAIN FACILITY" panel. If your control-instruments are already set (manually or previously) at the desired values, and there is no need to reset them again, press "Reject Changes?" and program will skip to the "NIPER MAIN FACILITY" without resetting the instruments.

If this is your first run and you have not previously set the default values:

(a) Set all the values on "NIPER Indicator" panel according to your experimental set-up.

(b) To save this setting as the new default setting, click on "Operate" in the pull-down menu bar and select "Make current values default."

(c) Press the "Accept Changes?" or the "Reject Changes?" button depending upon whether or not you are satisfied with the changes. If a dialogue box asks you whether you want to save changes, press "Yes."

(d) A second panel "NIPER Control" will open up. Follow steps (a) and (b) again.

(e) Press the "Accept Changes?" button if the control-instruments are to be reset with new values. Press "Reject Changes?" if no resetting of any of the instruments is needed. Press "Yes" in the dialogue box that may appear.

To Make a Run with a Previously Saved Setting

Previously stored settings cannot be used during run time for safety reasons. The procedure for selecting a previously stored setting involves opening the front panels of "NIPER Indicator" and "NIPER Control" VI's, selecting the desired setting using the log function, making them the default values, and closing the files. The step-by-step procedure is as follows:

1. Open "NIPER Indicator" or "NIPER Control" VI. These VI's can be opened directly from the finder using the "Find File" commands, or can be opened as follows:
   (a) Open and run "NIPER MAIN FACILITY."
   (b) When the "NIPER Indicator" front panel is opened, press "Reject Changes?" button.
   (c) When the "NIPER Control" front panel is opened, press "Reject Changes?" button.
   (d) Press the "LOG CONTROL" button on the "NIPER MAIN FACILITY" panel. Two dialogue boxes will appear asking questions. Properly answering these questions will open the desired front panels.
2. Select the "Retrieve Data" line from the pull-down "File" Menu. This will change the icons on the title bar.
3. Use the Up/Down Arrows at the title bar to select the desired Run.
4. Press the Eye Icon at the title bar to view the settings of the run number selected in step 3 above. The date and time of setting also appear on the title bar. Use "Name of Setting" to identify the setting. If not satisfied with selection, repeat steps 3 and 4 until satisfied.
5. Select "Make Current Values Default" from the pull-down "Operate" menu.
6. Close the file ("NIPER Indicator" and/or "NIPER Control" panel VI's).

Result: This setting will be used in all the following runs until a new default is selected or settings are changed during a test.

To Adjust Instrument Settings During Run Time and Save it Later

1. Press the "ADJUST CONTROL" button from the control bar. It will open up "NIPER Indicator" panel.
2. Make changes as desired.
3. To save this setting as the new default setting, select "Operate" in the pull-down menu bar and then select "Make current values default". This step may be skipped if the current changes are only temporary (i.e., for the current run only).
4. Press the "Accept Changes?" or the "Reject Changes?" button depending upon whether or not you are satisfied with the changes. If a dialogue box asks you whether you want to save changes, press "Yes."
5. A second panel "NIPER Control" will open up. Follow steps 2 through 4 again. If "Accept Changes?" button is pressed on the "NIPER Control" panel, the program will begin to reset the control-instruments with these new values. If "Reject Changes?" is pressed, the program will ignore the changes and will not reset any instruments.

To Store Settings in the Library

The settings cannot be stored during run time for safety reasons. The procedure, therefore, involves opening the front panels of "NIPER Indicator" and "NIPER Control" VI's, providing necessary information, and saving the desired settings using log function, selecting one of the new settings as default (if desired), and closing the files.

1. Open "NIPER Indicator" and/or "NIPER Control Panel" VI. These VI's can be opened directly from the finder using the "Find File" functions, or can be opened as follows:
   (a) Run "NIPER MAIN FACILITY".
   (b) When "NIPER Indicator" front panel is opened, press "Reject Changes?" button.
   (c) When the "NIPER Control" front panel is opened, press "Reject Changes?" button.
   (d) Press the "LOG CONTROL" button on the "NIPER MAIN FACILITY." Two dialogue boxes will appear asking questions. Properly answering these questions will open the desired front panels.
2. Enter the needed information for the set-up. Do not forget to enter a suitable name for the setting in the box "IDENTIFYING NAME OF THIS....."

3. Select the "Log Front Panel" line from the pull-down File Menu. This will store the current setting, the current date and time, and assign a run number. Store as many settings as desired following steps 2-3.
4. Select one of the new settings as a default if desired. Otherwise, the previous default values will prevail.
5. Close the file ("NIPER Indicator" and/or "NIPER Control Panel" VI's).

   Result: The logged settings will be stored and can be used in the following runs as described in the section "To Make a Run with a Previously Saved Setting.".

Installing The Driver to the Automation Program

1. By using the "Find" command under the menu option "File" (or using ⌘+F) in the "Finder" mode, locate the "Instrument Drivers" folder and place the drivers in it. Remember the name of drivers and their current locations. This step is optional, but it helps to keep all drivers in one folder for future reference.
2. Locate and open the "Instrument VI's" folder. In the "Instrument VI's" folder, open the file "Indicator Driver Selector" or the file "Control Driver Selector," depending upon the type of driver, i.e., whether it is an indicator driver or a control driver. If not sure, see "To Classify Instruments On the Basis of Their Functionality."
3. Under the menu option "Windows," select the "Show Diagram" command.
4. The box, with the number and arrows at the top of the border, is called the case box. For further information on the case box, refer to the instructions provided on the VI, or LabVIEW 2 User's Manual. Using the arrows, go to the last case; i.e., the case showing the highest number.
5. Point the mouse anywhere along the border of the case box. Hold the command key along with the mouse button (command-click). A "Pop Up Menu" will appear.
6. Select the command "Add Case After." An empty case window is now created to hold the new driver that is to be installed.
7. Command click on the empty window. From the "Pop Up Menu" select "VI."
8. Select the Mettler PJ-15 driver to be installed. The name and location of the driver were noted in step 1.
9. Under the menu option "Tools," select the wiring tool (the spool of wire located in the middle of the second column). The National Instruments "Getting Started Manual" (1991) explains how the wiring tool works. Wire the top left of the driver icon to the Indicator Panel frame (to the black spot on the bottom of the frame); wire the top right of the driver icon to the Drivers Outputs (to the black spot on the right side of the frame). See previous frames to use as examples.

10. Note the case number of the window in which the driver was just installed; the case number now corresponds to the driver number. Suppose the case number of the window containing the PJ-15 driver is 5.
11. Save this set-up.
12. Now the driver is ready to allow communication between the computer and the instrument.

To Classify Instruments On The Basis Of Their Functionality

Instruments can be classified as follows:

Indicator instruments. Indicator instruments send signals to the computer. They do not respond to commands from the computer. Examples of indicator instruments are thermocouples, pressure sensors, etc.

Control instruments. Control instruments receive signals from the computer. They respond to the commands from the computer and change their status accordingly, e.g. turn themselves ON or OFF, change their pumping rate, etc.

Dual-Function instruments. Dual-function instruments both send signals to the computer and receive signals from the computer. They send signals to the computer when queried, as well as respond to the commands from the computer and change their status accordingly. Thus, they act both like an indicator instrument and a control instrument. One example of dual-function instruments is a Mettler's PJ-15 balance which sends weight reading (an indicator function) when asked by the computer, and resets its "TARE" or turns ON/OFF when commanded by the computer. Another example is a Scanivalve (scanning valve instrument), which expects the computer to tell it which channel's reading is desired, positions itself to the commanded channel (a control function), and then sends the reading (an indicator function) of that channel to the computer. The dual-function instruments require two separate drivers, one compatible with indicator driver's format and the other with control driver's format. Although dual-function instruments provide both functions, it is not required that both functions be used. Thus, either function can be used alone. In the Mettler PJ-15 example above, it is OK to use the dual-function instrument for reading the weight only and not worry about "TARE," or vice versa.

To Analyze A System Configuration Problem

The worksheet in Table 4.1 can be used to analyze a configuration problem See problem sets in chapter 9 for example use of this worksheet.

TABLE 4.1

Problem Analysis Work Sheet

STEP 1: Enlist All Automation Instruments In Three Categories (See ***To Classify Instruments On The Basis Of Their Functionality*** for more details):

Indicator instruments:	
Control instruments:	
Dual-Function instruments:	

STEP 2: All automation instruments to be reset at start-up (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above):

STEP 3: All automation instruments to be reset in case of emergency (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above):

STEP 4: All automation instruments to be reset at run shut-off (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above):

STEP 5: All automation instruments to be reset in case of warning (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above):

WARNING CONDITIONS	CONTROL INSTRUMENTS	RESPONSE

STEP 6: Enlist All Automation Instruments Whose Operation Is Subjected To Meeting Constraints With Object Instrument(s) (i.e. the control instruments which can be operated only when the status of target instrument(s) meets the constraint conditions):

CONSTRAINED INSTRUMENT	OBJECT INSTRUMENT	LOGICAL CONSTRAINT

To Determine Correct Parameters To Be Entered Into The NIPER INDICATOR VI

The worksheet in Table 4.2 can be used to organize data to be entered into the front panel of the NIPER INDICATOR VI when it opens as the NIPER MAIN FACILITY is run. See problem sets in chapter 9 for example use of this worksheet.

To Determine Correct Parameters To Be Entered Into The NIPER CONTROL VI

The worksheet in Table 4.3 can be used to organize data to be entered into the front panel of the NIPER CONTROL VI when it opens as the NIPER MAIN FACILITY is run. See problem sets in chapter 9 for example use of this worksheet.

To Configure NIPER MAIN FACILITY for Specific Automation Setups

The following general steps can be used to configure the NIPER MAIN FACILITY for most automation requirements (See example use of these steps in example problems of Chapter 9):

1. Using Table 4.1, analyze the system configuration.
2. Organize data to be entered into the front panel of the NIPER INDICATOR VI. Use Table 4.2 as template to facilitate this task. Reviewing Table 4.1 carefully helps in accomplishing this task because this table is essentially a transformation of the system configuration to a format readily feedable to the NIPER INDICATOR VI.
3. Organize data to be entered into the front panel of the NIPER CONTROL VI. Use Table 4.3 as template to facilitate this task. Again, reviewing Table 4.1 helps in accomplishing this task because this table is essentially a transformation of the system configuration to a format readily feedable to the NIPER CONTROL VI.
4. Locate suitable driver for each item and make a list of their names and locations. Table 4.4 can be used as example.
5. Load all drivers in the "Indicator Driver Selector" VI or in the "Control Driver Selector" VI depending upon whether the instrument is an indicator or a control (i.e. whether it is listed in the worksheet of step 2 or step 3). The section ***"To Install a Driver Into the NIPER Lab WARDEN Program"*** contains more details on how to install a driver. Load every driver in its correct case. For example, if a driver has been designated as driver 0 in Table 4.2 (row 8, col. 2) or Table 4.3 (row 9, col. 2), it should be loaded in case 0. If a case already has a driver, the existing driver can be replaced with the new driver. However, if previous drivers are desired to stay loaded, load the driver into the last case, and consequently, re-designate the driver number in Table 4.2 or Table 4.3.

TABLE 4.2

NIPER Indicator Panel Settings

BOX NAME	Entry For Set #0	Entry For Set #1
Frame No.		
ITEM NAME		
Item No.		
Communication Information		
Board No.		
Channel No.		
Driver VI		
EMERGENCY & WARNING LIMITS		
Cutoff Higher Limit		
Cutoff Lower Limit		
Cutoff Accel. (% per min)		
Cutoff Decel. (% per min)		
Warning Higher Limit		
Warning Lower Limit		
Warning Accel. (% per min)		
Warning Decel. (% per min)		
IDENTIFYING NAME OF THESE SETTINGS		
COMMON CONFIGURATION (ALL ITEMS)		
Interval (seconds)		
Test Status		
Multiplex Boards		
ACTIONS DESIRED TO CORRECT WARN. COND.		
FRAME NO = 0		
Responding Instrument		
Lower or Higher Limit Status		
Reset at % of Value		
FRAME NO = 1		
Responding Instrument		
Lower or Higher Limit Status		
Reset at % of Value		

TABLE 4.3

NIPER Control Panel Settings

BOX NAME	Frame 0 Entry	Frame 1 Entry	Frame 2 Entry
Frame No.			
ITEM NAME			
Item No.			
SET VALUE			
COMMUNICATION INFORMATION			
Board No.			
Channel No.			
Driver VI			
EMERGENCY RESPONSE INFO			
Emerg. Sequence Position			
Emerg. Value to be set			
SYSTEM SHUTOFF INFO			
Shutoff Sequence Position			
Value to be set at Shutoff			
IDENTIFYING NAME OF THESE SETTINGS			
CONSTRAINTS (If none, empty array)			
FRAME NO = 0			
Logical Condition (No name on box)			
Item on frame number			
FRAME NO = 1			
Logical Condition (No name on box)			
Item on frame number			
FRAME NO = 2			
Logical Condition (No name on box)			
Item on frame number			

TABLE 4.4

List of Drivers and Directory Location(s)

Instrument	Driver Name	Directory Location
Balance	DRIVER RS232 Mettler PJ-15	La Cie 200-Q:NIPER Lab WARDEN: Readout & Control Facility: Instrument Drivers:
Timer	DRIVER Timer Example	ditto
Valve	DRIVER Solenoid ON/OFF EXAMPL	ditto
Pump	DRIVER Solenoid ON/OFF EXAMPL	ditto

For example, as the driver 0 in Table 4.2 was being loaded into case 0, it was decided to leave the existing driver in case 0 for future use. Instead, the driver 0 was loaded into case 7 (case 0 through 6 already had drivers). Thus, the driver designation in Table 4.2 had to be changed from driver 0 to driver 7.

6. Open and run NIPER Lab WARDEN from the pull-down apple menu or by using "Find File" functions (⌘+F).
7. Open and run NIPER MAIN FACILITY. Use of NIPER Lab WARDEN to open this facility is more convenient than using "Find File" functions (⌘+F). As the program is run, the front panel of NIPER INDICATOR VI will open up.
8. Enter information from Table 4.2 (worksheet for indicator instruments) into this panel. Be aware that some boxes have more than one frames. Use arrows to select the right frame number before entering information for that frame. If you want to save it for future use, then select "Make Current Values Default" from the pull-down "Operate" menu. Finally, press Accept Current Settings? to use these settings in the current run. A dialogue box may appear asking if you want to save changes. Respond as desired.
9. The front panel of NIPER CONTROL VI will open up next. Do the same as in step 8 except that use Table 4.3 (worksheet for control instruments) instead of Table 4.2.

TROUBLE SHOOTING

SYMPTOM No. 1: LabVIEW quits because of memory shortage, or the icon frequently changes to a tractor during execution.

POSSIBLE REMEDIES: Allocate more memory to LabVIEW by taking these steps:

1. Quit LabVIEW application. Be sure to save all the files desired.
2. Go to Finder. This can be done by clicking on an empty spot in screen, or selecting "Finder" from the pull-down menu under the top-right icon.
3. Find and select "LabVIEW" application program.

4. Select the "Get Info" line from the pull-down "File" menu under the "Finder" application or press "⌘" and "I" simultaneously. This will open a screen showing the suggested size and current size of the memory.
5. Increase the current size as much as possible to reduce memory shortage problems. However, remember that allocating more memory to LabVIEW will decrease the memory available for other applications. Thus, if you want to review your data in other applications such as Microsoft Word or Excel while LabVIEW is running, you may have to assign optimum memory to each application. Experience will eventually help you in determining the optimum memory size to solve this problem.

SYMPTOM No. 2: The program does not behave as described in the manual.

POSSIBLE REMEDIES: This is likely to happen when the user has taken out-of-sequence actions. There is no harm in doing that, and with a little experience, the user will be able to continue on from where he/she left off. A few simple remedies that may solve the problem, in order of increasing probability are:

1. Re-run the program.
2. Close the program, re-open it, and run it again.
3. Quit the LabVIEW application, re-open your program, and run it again.
4. Re-start the computer and run the program.

SYMPTOM No. 3: When "NIPER MAIN FACILITY" is running and the front panels of "NIPER Indicator" and "NIPER Control" VI's are opened, the log functions under the pull-down "File" menu are disabled (dimmed).

POSSIBLE REMEDIES: The log functions are not available during the run time. The only way to save or retrieve settings to/from logs is to do it prior to or after the program execution. To save settings that you made during run time, select "Make Current Values Default" from the pull-down "Operate" menu, answer "Yes" in the dialogue box to save, and continue with the test. After the run is stopped, but before the "NIPER MAIN FACILITY" panel is closed, open the front panels "NIPER Control" and/or "NIPER Indicator." Step 1 of the section titled "To Store Settings in the Library" can be used to open these panels conveniently, and log the settings. To use one of the previous settings for the current run, open the front panel(s) "NIPER Control" and/or "NIPER Indicator," select the setting from the log, make it default, and close the file. Run the "NIPER MAIN FACILITY". It will show the setting as the default. Press "Accept Changes?" without making any adjustments.

Chapter 5

DESCRIPTION OF NIPER DISPLAY FACILITY

INTRODUCTION

Features

The "NIPER DISPLAY FACILITY" allows the user to examine the current run in progress and has the following attributes:

(1) Gives a pictorial representation of the physical configuration of the process being controlled,

(2) Examines data in numeric form,

(3) Reads error messages,

(4) Examines the status of the test with regard to warning and emergency conditions, and

(5) Examines the instruments being adjusted by the computer to overcome warning conditions and their reset values.

In essence, this facility provides a pictorial view of the data and status of the tests currently in progress. Its pictorial representation helps to identify potential problems and assess their urgency, and therefore is the facility of choice when a test is in progress and operator is not fully attentive. It is an optional facility that need not run for reading/controlling/monitoring the instruments or graphic display of the data. It only displays data when "NIPER MAIN FACILITY" is running in the background.

Technical Information for Programmers

This program reads data to be displayed from "NIPER GLOBAL Data" VI. This VI is initialized and updated every time new data is scanned by "NIPER MAIN FACILITY" program. The data received is sorted according to each item's number designated by the user in the "NIPER Indicator" VI. This sorted array is then indexed to update the displays in sequential order. The advantage of this strategy is that the instrument need not be in order and can be positioned anywhere on the "NIPER Indicator" box. Therefore, instruments can be added or removed easily.

Because of the sequential handling of the data, the following are prerequisites for complete and proper functioning: (1) The number of active items (i.e., the ones with item #≥0) in the "NIPER Indicator" VI must be equal to the number of items on the front panel of this facility. If not, there will not be enough data to update all items or vice versa (i.e., there will not be enough items to display all data); (2) No two items on "NIPER Indicator" VI should have the same item number. The entire ordering after each set of duplicate item numbers will be offset by one.

It must be emphasized that the process diagram shown in Fig. 2.4 (the front panel for "NIPER DISPLAY FACILITY") is not a universal diagram, nor is it necessary to have this facility. Fig. 2.4 was designed for a visual display of high temperature and pressure experiments in NIPER's steamflood laboratory. Users should build their own front panel by copying the indicators from LabVIEW's built-in libraries, or other VIs available to the user. The "NIPER DISPLAY FACILITY" VI can be used as a template to draw/configure such a diagram. After drawing such diagram and designating each item in the diagram, a unique number (is increment of 1, starting from zero), their terminals have to be repositioned within the diagram (i.e., the terminals have to be dragged into the frames of corresponding case numbers). The process is rather simple and explained in Chapter 6. National Instruments operating/programming manuals also provide information on how to build indicator (display items)..

DESCRIPTION OF DISPLAYS IN "NIPER DISPLAY FACILITY"

Controls and Displays

In the "NIPER DISPLAY FACILITY" (Fig. 2.4) some of the features that have been incorporated include:

"STOP BUTTON" stops the execution of this VI and lets user activate another VI without unloading the program. Use this button instead of "close file" options when this VI will be used again in the session to avoid unnecessary loading and unloading.

"DATA & MESSAGES RECEIVED" box contains a set of indicators for each item that display the item's name and number, its current value, and error messages. It also displays the user-specified cutoffs (higher-limit and/or lower limit) for the item for comparison with its current value. Information about the desired item can be read by using the arrows on the box or typing in the frame number on which the item is located.

"ITEMS REQUIRING PRE-EMERGENCY ADJUSTMENTS" box contains information about the corrective actions being taken by the "NIPER MAIN FACILITY" during a warning situation. Each frame of this box contains the item number and the percentage change to be made to the item's current value. Zero percent implies the item is being turned OFF and 999% implies it is being turned ON. Using arrows on the box, or typing in a frame number, all the items requiring adjustments can be seen.

"CHANNELS WITH WARNING AND/OR EMERGENCY CONDITIONS" is a table which provides a summary of the status of the test. The names of items with warning or emergency conditions are displayed along with item number, error messages, and values in appropriate columns. If no such conditions exist, the table is blank. The scroll windows can be used to browse through the table.

"EMERGENCY?" indicator lights up when one or more items are encountering an emergency situation. It is recommended that the VI be closed and "NIPER MAIN FACILITY" be activated when this button lights up.

"CUSTOM DISPLAYS" are created by the user in the process of building the process diagram. The user should keep adequate documentation for all the elements in his diagram. An easy way to keep a description of an item is to package it along with the item itself, as described in the next chapter. The custom displays shown in "NIPER DISPLAY FACILITY" (Fig. 2.4) depict various elements of a high-temperature, high-pressure experimental laboratory.

(This page intentionally left blank)

Chapter 6

OPERATION OF NIPER DISPLAY FACILITY

CUSTOM DESIGNING A DISPLAY PANEL IN NIPER DISPLAY FACILITY

The user needs to custom-make this facility according to his/her system set-up (instrumentation). Before undertaking this task, one needs to ascertain its applicability because it is only an optional facility whose sole purpose is to present data pictorially. For systems to be used extensively or for long durations, building this facility may be a worthwhile investment. The following step-by-step procedure can be followed to build a display facility.

Once the facility is built, it does not require much user input. All the user needs to know is how to use the "stop" button, scroll through the table, and browse through boxes—techniques already mastered by the user at this point.

1. Open a new Canvas file, make a complete drawing of the process, and save a copy of it. This copy will be referred to later as "original drawing."

   Canvas or an application of your choice can be used (if it allows various elements of the drawing to be grouped/ungrouped, and allows the sketch to be saved in "PICT" format). The drawings should have a resemblance to the actual items or at least be recognizable. Time can be saved by copying and converting various sections of the drawings from clip-art libraries (which also come with many applications) into active LabVIEW indicator displays. Even more time can be saved if an active LabVIEW display is copied from LabVIEW VIs and pasted directly without any conversion. The folder "Sample Displays" includes several files containing drawings and active LabVIEW displays of commonly-used instruments, e.g., a pump.
2. Start from a copy of the original drawing. Group elements of this drawing into distinct items. Sort them according to their usage into three categories: Boolean (i.e., buttons of displays which show on/off status such as valves); numerics (those whose range can be represented pictorially such as an oven with possibly three conditions: normal, under-, and over-limit); and inactives (items which do not change configurations, such as pipes etc.).
3. Make an additional sketch for each Boolean. The pair of sketches for each Boolean may be similar in shape and differ only in color, or may be a different drawing altogether.
4. Make additional sketches for each numeric depending on how many conditions they represent, e.g., three sketches will be needed for the oven example mentioned above. Again, they may be the same sketches with different colors or may be different

drawings. Save and close the file. This file will be referred to later as "parts drawings."

5. Open the "DISPLAY Template."
6. Stop the execution by pressing the oval in the control bar if the program is running (or if the arrow on the control bar is darkened).
7. Save a copy of "DISPLAY Template" with a different name of your choice. This may be done by selecting "Save As ..." from the pull-down "File" menu. A suggested name is "My Company DISPLAY FACILITY." Save it in the same folder containing "NIPER Lab WARDEN."
8. Open the "original drawing" file, and copy the entire diagram on the clipboard.
9. Activate "My company DISPLAY FACILITY" and paste the drawing from the clipboard on to its front panel.
10. Select "Numeric" from the pull-down "Controls" menu and select "Custom PICT Ring" from the extended menu. A display will appear with a blank label. Type "temp-numeric" as the label and press enter.
11. Select "Boolean" from the pull-down "Controls" menu and select "Custom PICT Boolean" from the extended menu. A display will appear with a blank label. Type "temp-Boolean" as the label and press enter.
12. Inspect your diagram in some sequence. Select the first active item and convert it to a LabVIEW display (Step 13) and package its description with the item (Step 14).. Skip non-active items because they do not need to be modified in LabVIEW.
13. To convert an active item to a LabVIEW display: Start numbering the item from 00, 01, 02, etc.

    First try to copy similar LabVIEW active displays from other VI's. If such a copy is found, then position it over the selected item in the drawing. Then skip to part "f."

    a. Make a copy of the "temp-Boolean" or "temp-numeric" and change its name to the name and sequence number of the selected item, e.g., "00-Steam Feeder Tank." Copies can be made by selecting an item and dragging it while the option key is pressed. Position this over the selected item in the drawing.
    b. Open or activate the "parts drawings" file. Select the picture corresponding to the item being converted in LabVIEW. There may be more than one picture for the item, depicting various states, e.g., "ON" and "OFF," or "LOW," "MEDIUM," AND "HIGH." Select and copy the picture that corresponds to the desired state.
    c. Open or activate "My Company DISPLAY FACILITY."

d. Select the "Custom PICT Boolean" or "Custom PICT Numeric" display positioned over item 1 (the item being converted to LabVIEW). Click on the display to show the right frame, i.e., "PICT OFF," "PICT ON," OR "PICT 1." Paste the picture from the clipboard onto the frame.

e. Repeat steps b through d until pictures have been pasted into all positions of the display.

f. Change the display to an indicator (if it is not already an indicator) by pointing to the item while keeping the ⌘ (command) key pressed and selecting "Change To Indicator."

14. To package information with an item:
    a. Point the cursor to the item. Cursor may be anyone of the six tools except the brush.
    b. Press ⌘ and click-hold. The cursor will change to an arrow.
    c. Select "Description..." from the pop-up menu.
    d. Type in the description about the item in the box that appears.
    e. Press "OK" when finished. The box will disappear, and the information will be saved, if you save the file before closing it.
15. Select the next active item. Convert it to LabVIEW display as in step 13 and step 14 in these instructions.
16. Repeat step 15 until all active items have been converted to LabVIEW displays.
17. Select "Show Diagram" under pull-down "Windows" menu [or press ⌘ (command)+F)]. This will open the block diagram of "My Company DISPLAY FACILITY" VI.
18. Drag terminals (small rectangles with labels) into the box labeled "CASE SELECTOR FOR DISPLAYS." Do it sequentially and place each terminal into its corresponding case number. For example, the terminal for item number 6 has to be placed in the case whose frame number is "6." Wire this to the not-equal node if it is a Boolean. If it is a numeric, wire it to the Comparison-Select node and enter the upper and lower range values in the boxes showing 500 and 100. Follow the example shown on the block diagram.
19. Double-check to make sure there is a one-to-one correspondence between frame numbers and item numbers (they have to be identical). Also make certain that every terminal has been placed in a Selector, and there is no frame in the box.
20. Save the program. It is now ready to operate.

TROUBLE SHOOTING TIPS FOR A CUSTOM-MADE DISPLAY FACILITY

The following are helpful hints that may be useful in trouble-shooting custom-made display facilities.

SYMPTOM No. 1: Nothing seems to happen when the program is run; i.e., all the displays remain unchanged, along with the table and boxes.

POSSIBLE REMEDIES: The most likely cause of the problem is that "NIPER MAIN FACILITY" is probably not yet running, or its "DATA UPDATE" button is not pressed. Less likely causes are that it has not been set to acquire data, or has not acquired a new set of readings after a custom-made display facility was loaded (opened). First, load or reactivate the "NIPER MAIN FACILITY" panel and make sure it is running and its "DATA UPDATE" button is pressed, and then press the "GET NEW DATA" button to acquire current data. Also, check if the parameters have been properly set by pressing the "ADJUST CONTROL" button and re-examining the set-up on "NIPER Indicator" panel.

SYMPTOM No. 2: Some displays do not seem to change at all. Is this OK?

POSSIBLE REMEDIES: No. All displays should be updated sequentially even if the value remains the same. Therefore, such a condition indicates improper control settings in the "NIPER Indicator" VI.

The probable causes are: (a) The number of items in the "NIPER Indicator" VI setup is not equal to the number of items on the front panel of the custom-made display facility. Thus, there is not enough data to update all items or vice versa (i.e., there are not enough items to display all data), (b) two or more items on "NIPER Indicator" VI have the same item numbers. Therefore, the ordering after a set of duplicate item numbers has been offset by one. The easiest solution is to activate the "NIPER MAIN FACILITY" and re-adjust indicator settings.

If this does not solve the problem, re-examine the custom-made display facility's panel and block diagram. Specifically, check the following: (a) Each terminal is in the right frame (i.e., the item number and frame number are the same), (b) there is only one terminal in each frame, (c) each frame has a terminal. Additional empty frames are allowed only after the last frame containing a terminal (i.e., the one with the terminal of highest item number). For example, if there are three items in the set-up: item 1 would go to frame 1, item 2 would go to frame 2, item 3 would go to frame 3. There could be empty frames numbering 4, 5, 6, and so on. It is not permitted to have frame number 2 empty or to place item 1 in frame 3. These pitfalls would violate the one-to-one correspondence between frame number and item number.

SYMPTOM No. 3: Frequent breakdowns, freeze-ups, shut-downs, malfunctions, etc.

POSSIBLE REMEDIES: These symptoms are frequently caused by "Mad-Mouse Syndrome" wherein the mad-mouser chronically and habitually keeps moving, clicking or otherwise using his mouse. The result is an overloading of the service requests in the queue, omission of some requests, and frequent disruption of the CPU. Sadly, there is no known cure of this symptom. Only time will teach the user to be in sync with the software and have patience until they obtain a faster CPU.

(This page intentionally left blank)

Chapter 7

DESCRIPTION OF NIPER GRAPHIC FACILITY

A QUICK TOUR OF THE NIPER INTERACTIVE GRAPHICS

Introduction

The "NIPER GRAPHIC FACILITY" panel is shown in Fig. 7.1. The circled references for the icons are described in Table 7.1. This chapter describes the features of the "NIPER GRAPHIC FACILITY." The operational details for this facility are provided in chapter 8.

Features

This facility allows the user to:

(1) View the data from the current run or the previous runs,

(2) View the data in both numeric and graphical form,

(3) View only selected data from a run,

(4) Browse through previous data sequentially,

(5) View the smoothed data after curve fitting,

(6) Edit graphs in real-time; i.e. add, delete, copy points or segments, and

(7) Export data for viewing and processing in external applications such as Microsoft Excel.

It is an optional facility not required for interacting with the instruments. Therefore, the user does not need to run this facility for automation needs. A user can use it to browse, curve-fit, reduce, copy, and print previously logged data—and if a run is in progress, current data as well—at any time. The selected data can be easily edited by adding, deleting, or copying points or plot segments in real-time using mouse. The edited graph can be printed with LabVIEW print functions.

Technical Information for Programmers

This program reads data to be displayed from either a global data file (in tab-delimited "TEXT" format) or a log-file (in "DATA" format). These files are saved by "NIPER MAIN FACILITY." The format of log-data file does not allow it to be opened by most other applications besides LabVIEW.

The data received is processed before it is displayed. First, the data is reduced to the set of channels (items) that user has selected in "NIPER GRAPH CONFIGURATION" VI. This selected data is then curve-fitted according to the user-selected curve-fitting technique.

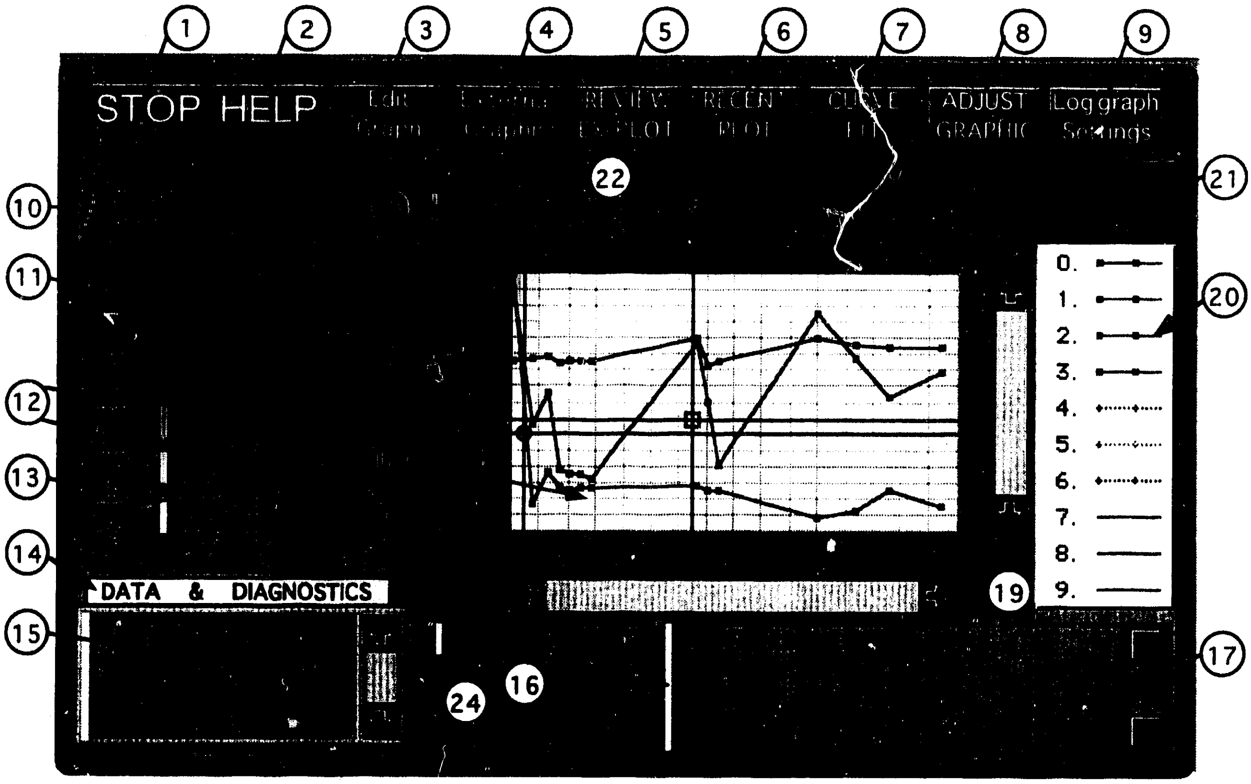


FIGURE. 7.1 - The front panel of "NIPER GRAPHIC FACILITY" VI.

TABLE 7.1
LEGEND FOR FIGURE 7.1

1. This button stops the run.
2. This button opens the "NIPER HELP" panel which provides information about LabVIEW and object-oriented programming. Like all of the other windows that are opened from the main panel, the "NIPER HELP" panel automatically disappears after a certain time duration, and the main panel then reappears.
3. This button magnifies the graph shown in Fig. 7.1 to a full size screen-editable graph. This feature provides a better view of the data being graphed, and allows real-time editing of the plot: i.e., points or segments can be added, deleted, and copied from this or other plots by just clicking and pointing the mouse. The edited plot can be saved and printed.
4. This button relays information to graphics programs outside of the main application (or LabVIEW). The particular program may be reselected if desired. The exported data can be automatically plotted and rotated.
5. This button allows the user to view plots of previous runs one at a time. The desired run number is chosen using the selector (11 in Fig. 7.1). The selected number is shown in the display (13 in Fig. 7.1). After the run number is chosen, pressing this button will display the plot of that particular run.
6. This button displays plots of the data scanned from a run currently in progress, provided that the "NIPER MAIN FACILITY" is running and has the "UPDATE GRAPH" button pressed. If there is no run in progress, or the "NIPER MAIN FACILITY" is not running or does not have "UPDATE GRAPH" button pressed, the graphic palette becomes blank and a warning message appears in box 16. The display is updated automatically each time data arrives.
7. This button provides a curve-fit (either linear, polynomial, or exponential) to the data (regression analysis selection made by "ADJUST GRAPHIC" button, described in No. 8). The curve-fit option must already be activated when the new data arrives in order to update and display the curve-fit to the data.
8. This button opens a panel that allows user to adjust various aspects of the graphic display of data. This includes the selection of items to be plotted, data duration, x-axis time base, y-axis unit, type and order of curve-fit, and the name and location of log-file from which to read logged data. Any changes made to these settings may be saved or canceled.
9. This button allows selection of one of the set of graph settings previously stored by the user. This is useful when the user is using the program with multiple test configurations and wants to switch frequently from one setting to another. Each series of settings can be given a name for quick identification. This button can also be used to store a new set of graphic settings.

TABLE 7.1—Continued
LEGEND FOR FIGURE 7.1

10. This display shows the total number of ex-runs that are currently saved in the selected log-file.
11. This digital control allows the user to choose a specific ex-run number for plotting on the graphical display. It is operated by clicking on the arrows to increase or decrease the number shown in the adjacent box.
12. These displays are used to give the date and time of the selected ex-run, or the current run.
13. This display identifies the run which is being displayed on the graph.
14. This display changes color from green to red when there is an error message.
15. This display box shows the current data values received from an active test in progress. It also warns the user with diagnostic messages of any values that are outside the set limits.
16. This display shows messages concerning graphics. The display indicates the channels that are currently being plotted and the channels that are set for plotting but inactive for data acquisition.
17. This scroll bar gives a view of all the information presented in the "GRAPHIC MESSAGES" box (16 in Fig. 7.1).
18. This display presents data from a current run or an ex-run in a graphic format. The scales along the X-axis and Y-axis can be altered manually (expanded or reduced) in order to present any part of the plot desired for viewing.
19. These scroll bars along the bottom and right hand sides of the graphic display are another way the user can view parts of the plot not currently shown. However, using these scrolls does not rescale the graph. They merely unveil the hidden parts of the graph.
20. This is an indirect legend to the curves on the plot (No. 18). These color-coded lines determine the sequence number of various curves on the plot. The sequence number relates in ascending order, to the item selected for display. For example, if item number 2, 5 and 9 are selected for plotting, then the curve on the plot matching the legend line 0 will correspond to the item number 2; the legend line 1 will correspond to the item number 5, and the legend line 2 will correspond to the item number 9.
21. These displays both indicate and control the locations of the two cursors and their relative displacement from one another. If a cursor is moved by dragging, the current location of that cursor is automatically updated here to show accurate data values from the plot. Or, the desired cursor location (coordinates) can be typed in here, which will move the cursor to the typed-in position. The part of the plot displayed will change to follow the moving cursor. The X-column values indicate the X-coordinate locations of the cursors on the plot. The Y-column values indicate their Y-coordinate locations. The ΔX value indicates the horizontal displacement between the two cursors, and the ΔY value indicates their vertical displacement from one another.
22. These buttons are used to move the two cursors around the graph by small incremental distances. The part of the plot displayed will change to follow the moving cursor. The circle and square symbols in front of the buttons represent the two cursors, and the arrows on the buttons indicate the direction of movement on the plot. These buttons are particularly useful when jumping from one point on a curve to another directly, which happens when the cursor is locked over a point.
23. This display provides unit information (temperature, pressure, etc.) for the Y-axis.
24. This display provides the unit information (seconds, minutes, hours, etc.) for the X-axis.

The processed data is displayed on the graphic palette of the panel or exported according to the user's choice. The exportation of data is carried out by storing the processed data in a temporary file. A file of an external application such as Microsoft Excel is then opened. User has to write a macro or script to accomplish these functions. An example macro in Microsoft Excel is provided to help facilitate this task. This macro should be auto-executable so that it can perform the tasks associated with the external application upon its opening. This macro should call LabVIEW just before finishing the execution.

Menu Buttons

Located across the top of the "NIPER GRAPHIC FACILITY" panel, Fig. 7.1, are menu buttons that control various features of the program's operation. Each button is herein discussed.

STOP Button

The STOP button (1) allows the graphic facility to stop execution after completing the current cycle. This button is different than stopping execution by pressing the hexagon in the control palette, selecting "Abort" from the pull-down "Operate" menu, pressing '⌘+.', or closing the file—all of which terminate the program immediately. It is, therefore, recommended to stop execution by using this button in order to avoid any unlikely side-effects.

HELP Button

The HELP button (2) opens a panel titled "NIPER HELP" (Fig. 7.2), which provides information about LabVIEW and object-oriented programming. Like all of the panels that are opened from the main panel, "NIPER HELP" automatically disappears after a certain time duration. The main panel then reappears.

EDIT GRAPH Button

The EDIT GRAPH button (3) magnifies the graph shown in Fig. 7.1 to the graph shown in Fig. 7.3. This feature allows the data currently being displayed on the graphic palette to be displayed, printed, and saved in a full screen size format. The selected data can be easily edited by adding, deleting, or copying points or plot segments in real-time using a mouse. The edited graph can be printed with LabVIEW print functions and saved for future reference.

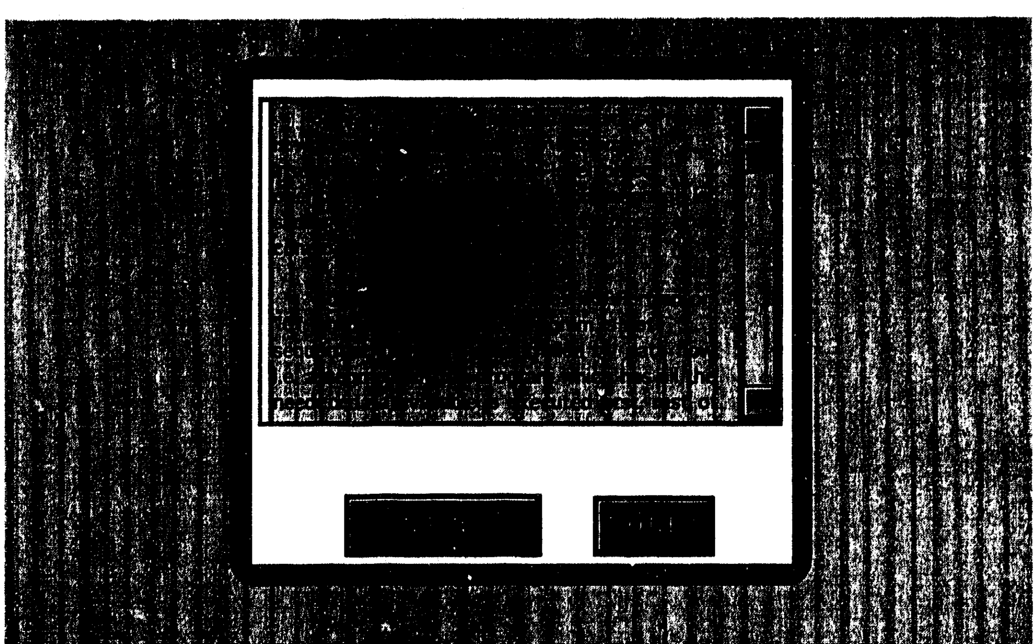


FIGURE. 7.2. - The front panel of "NIPER HELP" VI.

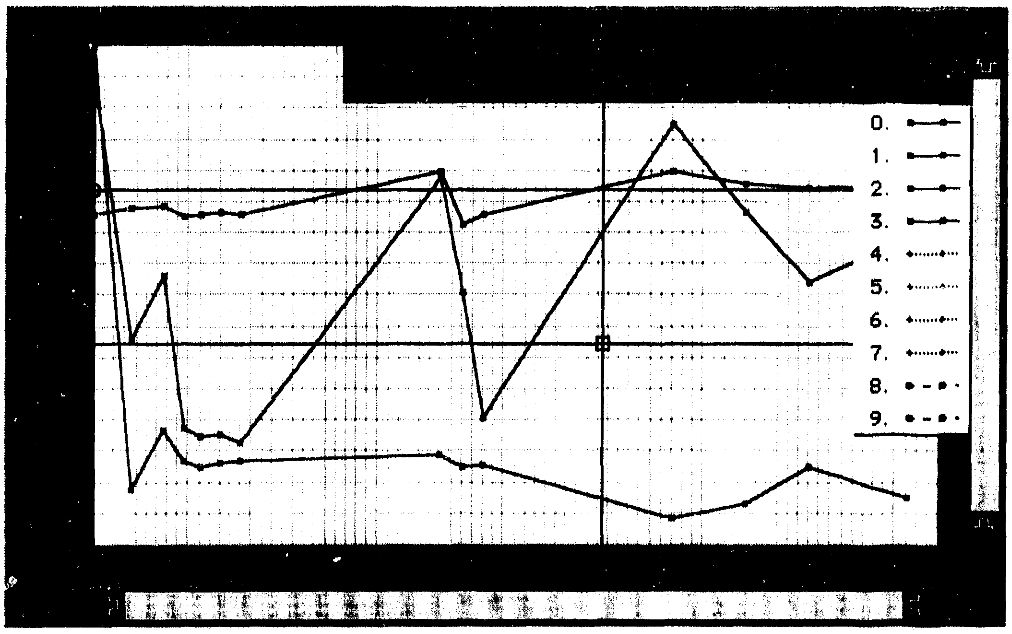


FIGURE 7.3 .- The front panel of "Full Size Graph" VI.

EXTERNAL GRAPHIC Button

The "EXTERNAL GRAPHIC" button (4) allows the data currently being displayed on the graph to be displayed and saved in an external application program.

REVIEW EX-PLOT Button

The "REVIEW EX-PLOT" button (5) allows you to view a selected plot of previous runs sequentially. After the run number is chosen, pressing this button will display the plot of that particular run.

RECENT PLOT Button

The "RECENT PLOT" button (6) displays the graph associated with the current run in progress. If no run is in progress at the time, the screen will become or stay blank.

CURVE FIT Button

This button (7) smoothes whatever data is selected prior to displaying it according to the type of curve-fit selected by the user.

ADJUST GRAPHIC Button

The "ADJUST GRAPHIC" button (8) opens a panel titled "NIPER GRAPHIC CONFIGURATION" as shown in Fig. 7.4. The circles in Fig. 7.4 are reference numbers that are described in Table 7.2. Through this panel, several data selection and graphic choices can be made such as the item numbers for which data is to be displayed, the name

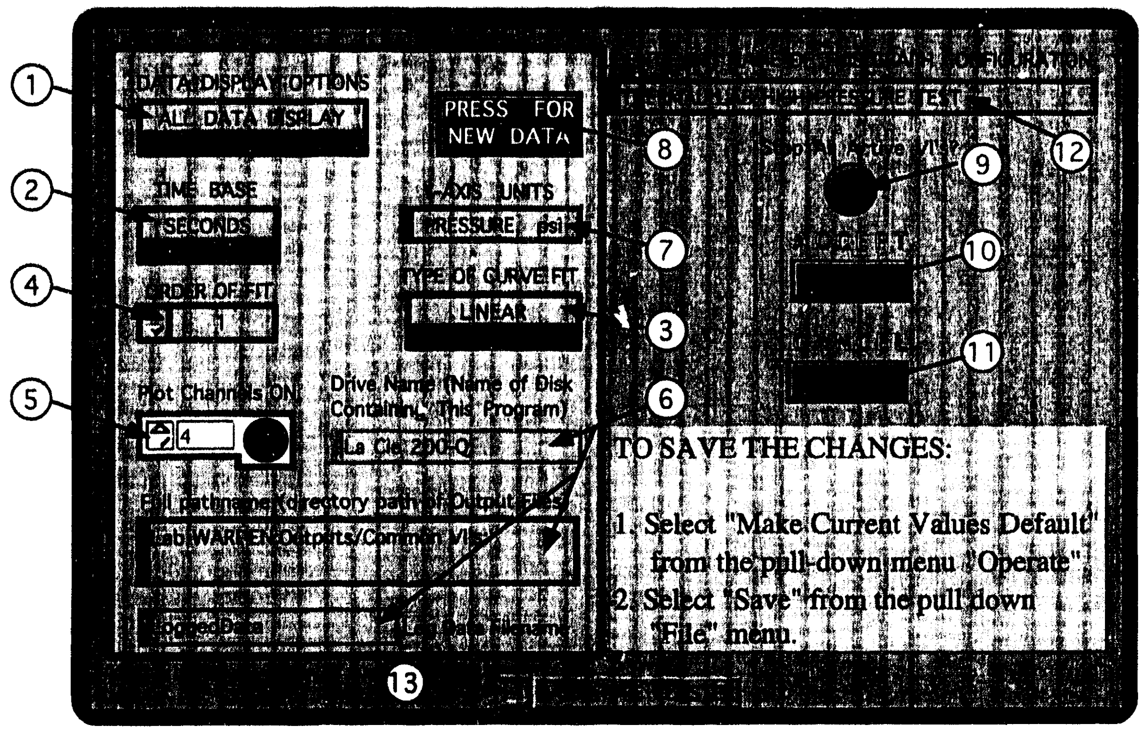


FIGURE. 7.4 - The front panel of "NIPER GRAPHIC CONFIGURATION" VI.

TABLE 7.2

LEGEND FOR FIGURE 7.4

1. This control allows plotting of only selected data. The options include no data display, all data display, new zero time, new append time. These functions are useful when multiple runs are carried out sequentially, or when a run has been interrupted for some reason and resumed later.
2. This control changes the time base of the plot in Fig. 7.1. The available time unit options include the following: HOURS, MINUTES, SECONDS (PORE VOLUMES and PICTORIAL are currently not available).
3. The "TYPE OF CURVE FIT" button is used to set the type of curve-fitting. Three curve-fit options are available: LINEAR, POLYNOMIAL, and EXPONENTIAL.
4. The "ORDER OF FIT" control selects the degree of the curve to be fitted. If either LINEAR or EXPONENTIAL is selected from REGRESSION ANALYSIS, the ORDER OF FIT setting is ignored. However, if the POLYNOMIAL option is selected, then the degree (order) of the polynomial fit needs to be set.
5. The "PLOT CHANNELS ON" button selects the channels (items) to be displayed on the plot located in Fig. 7.1. The circular button indicates the status of the currently selected channel. If the button is "in" (darker shade), the channel is active for graphical display. If the button is "out" (lighter shade), the selected channel is inactive.
6. These boxes describe the log-file's name and location (directory path name) for reading logged data. Several log-files can be stored and used selectively by using this function.
7. This box describes the units to be displayed on the Y-axis of the graph.
8. This button activates the selection made in the "DATA DISPLAY OPTION" box (#1 in Fig. 7.4).
9. All active VI's are stopped by this button. Useful for immediate termination of the run.
10. This button is used for accepting the changes for use in the current session.
11. Pressing this button will default to the original settings, thus nullifying the changes made during the current session.
12. This identifying name is useful in selecting this setting for future use.
13. This is an indicator version of the cluster containing boxes 1 through 8. It has been reduced in size to save the space, thus hiding the boxes.

of the log-file to be used during the current graphic session, the type and order of curve-fit desired, the time scale to be used (e.g. hours), and the X-axis units to be displayed (e.g., temperature and pressure). One could also select one of the previously saved graphic settings for the current session and/or for future sessions by making the selection as default. The identifying names (12 in Fig. 7.4) of the previously logged settings are helpful in locating a particular configuration.

LOG GRAPH SETTINGS Button

The "LOG GRAPH SETTINGS" button (9) opens the front panel of "NIPER GRAPH CONFIGURATION" (Fig. 7.4) and stops all VIs. This allows selection of one of the set of previously stored user defined graph settings. This button can also be used to store a new set of graphic settings. This button is useful when the user is using the program with multiple test configurations and wants to switch frequently from one setting to another. Each setting can be given a name for quick identification.

Run Number Displays

There are various displays on the "NIPER GRAPHIC FACILITY" that keep the user informed of different experimental conditions. The following displays that are discussed are all located on the "NIPER GRAPHIC FACILITIES" panel Fig. 7.1, unless specified otherwise.

NO. OF EX-RUNS Display

The display "NO. OF EX-RUNS" (10) shows the total number of ex-runs that are currently saved in the selected log-file.

RUN NO. Display

The "RUN NO." (11) control box allows the user to select the run number to be displayed when the "REVIEW EX-PLOT" button on the control bar is pressed. The value can be typed in or the arrows can be used to move up and down. For quick browse of data, these arrows can be used directly to review data sequentially. Any time a new input is selected on this control, the corresponding data is automatically displayed without pressing the "REVIEW EX-PLOT" button again.

RUN INFORMATION Display

This box displays the date (12), time (12), and run # (13) of the run being displayed on the graphics palette. If the data displayed is of the run currently in progress, it shows the current date, time, and the word "Current" in their respective indicators.

Information Displays

At the bottom of the main panel are three displays: "DATA & DIAGNOSTICS" label (14) changes color if an error message is received, a box below it (15), and "GRAPHIC MESSAGES" box (16). Along with the displays is an alarm that informs if there is a warning or error message. These display boxes provide various diagnostic messages about the experiment as described below.

DATA & DIAGNOSTICS ALARM

This indicator (14) changes color from green to red when a run currently in progress has an error or warning message. Like the previous box, it only functions when a run is currently in progress. When no run is in progress, it stays green.

DATA & DIAGNOSTICS Display

The "DATA & DIAGNOSTICS" display (15) shows the messages received from the "NIPER MAIN FACILITY" if a run is currently in progress. The messages include data, errors and their probable causes, and warnings. This box keeps the user informed of the current run status. It only displays information when a run is currently in progress, otherwise it stays blank.

GRAPHIC MESSAGES Display

The "GRAPHIC MESSAGES" display (16) gives information concerning graphics. The scrolling indicator box displays the item numbers of the data being displayed in the plot and other graphic error messages. When pertinent, it also suggests possible causes of any problems and remedial actions. These graphic messages pertain to the run selected, which could be the current run or one of the previously logged runs.

Graphical Display

In the center of the "NIPER GRAPHIC FACILITY" panel is the display (18) that presents data from the current runs or previous runs in a graphic format. The scale is automatically selected to show the entire plot every time new data is added; however, the numbers on the X-axis and Y-axis can be altered manually in order to present any part of the plot desired for viewing. The scales on the X and Y-axis can also be either expanded or reduced. The scroll bars along the bottom and right hand sides are another way the user can view parts of the plot not currently shown (19).

Graphic Display Legend

This legend (20) matches the colored lines on the plot with data channels activated for graphic display. However, the numbers in the legend do not correspond directly with the channel number. Instead, the numbers in the legend correspond to the index numbers of the series of selected channels arranged in ascending order and displayed in GRAPHIC MESSAGES display. For example, if channel 3, 7 and 11 are selected, line 0 in the legend will correspond to the lowest plotted channel listed in the GRAPHIC MESSAGES display, i.e., number 3. Line 1 in the legend will correspond to channel number 7, the second-lowest plotted channel listed in the display. Line 2 will corresponds to channel number 11, the third-lowest plotted channel listed in display.

Graphics Cursors

Two graphic cursors are located above the graphic display (23). The circle and square symbols left of the arrow buttons (23) represent the two cursors. The purpose of these cursors is to help obtain accurate data values from the plot.

Cursor Position

Located near the top of the graphic display (22) are windows that pertain to the cursors' position. Both the square and circle cursors share the X, Y, ΔX, and ΔY columns (22). The X column values indicate the X coordinate locations of the cursors. The Y column values indicate the Y coordinate locations of the cursors. The ΔX value indicates the horizontal displacement between the two cursors. The ΔY value indicates the vertical displacement between the two cursors. These windows are dual-purpose. They can be used to move cursor(s) to desired locations(s) by typing in coordinates, and they can be used to read accurate data when the cursor is placed over a point in the graph using mouse or arrows.

Cursor Movement

Located near the top of the graphic display are buttons (23) that pertain to the cursors' movement. The arrows on the buttons indicate the directions of movement of the cursor in the graphic display. As discussed in the above section "Cursor Position," the cursor can also be moved by typing in the coordinates.

Chapter 8

OPERATION OF NIPER GRAPHIC FACILITY

INTRODUCTION

Features

The features of the "NIPER GRAPHIC FACILITY" were described in chapter 7. This chapter describes how to operate the "NIPER GRAPHIC FACILITY" panel.

Starting Up the Program

Open the "NIPER GRAPHIC FACILITY" file. It can be opened conveniently using "NIPER Lab WARDEN" buttons, or using "Find File" functions and double-clicking. It will automatically start running, which is indicated by a darkened arrow on the control bar. If it is not running for some reason (i.e., the arrow is hollow), then click on the arrow to run it.

Upon start-up, the program will open up another front panel—"NIPER GRAPH CONFIGURATION." Select the desired options. Check the box "LOG DATA FILES FULL PATH NAME" and edit the path name if it needs corrections. The path name should point to a LabVIEW "log" file. Other types of files can not be opened by this facility. If satisfied with the changes, press "ACCEPT." Otherwise, press CANCEL. Either way, this front panel will be closed, and the "NIPER GRAPHIC FACILITY" panel will be activated.

To Get Information

On the "NIPER GRAPHIC FACILITY" panel (Fig. 7.1), click the mouse on the HELP button to obtain information about LabVIEW and object-oriented programming. Pressing this button opens up the "NIPER HELP" panel. This panel (Fig 7.3) contains a display, scroll bar, and two buttons. The display contains a brief text explaining some of the key concepts about object-oriented programming and the use of the LabVIEW software. The entire text can be viewed by using the scroll bar located along the right border of the box. One of the buttons below the display is labeled CONTINUE. This button is provided because this "NIPER HELP" panel remains open only for roughly a minute. "NIPER HELP" automatically exits back to the "NIPER GRAPHIC FACILITY" panel after one minute. Click the CONTINUE button to receive an additional 60 seconds of time for reading through the help information. Click on the DONE button to immediately exit back to the "NIPER GRAPHIC FACILITY" panel.

To View Data and Diagnostic Messages

The display labeled DATA AND DIAGNOSTICS, located in the lower left corner of the "NIPER GRAPHIC FACILITY" panel (button #15 in Fig. 7.1), provides information about a test which may be in progress. Indicator instrument values and diagnostic messages are received here from "NIPER MAIN FACILITY", if it is running. The display box presents the most recent readings from those instruments activated for data acquisition. The display lists the item numbers in ascending order along with their corresponding data values, including units. Along with the data values for each item, a diagnostic message appears if there is one. This message is a warning indicating that the data value for that channel is outside of the user-defined limits. Check the warning message carefully to determine if there is a dangerous problem developing. If so, close the "NIPER GRAPHIC FACILITY" VI and return to "NIPER MAIN FACILITY."

To Select the Channels for Plotting

Press the ADJUST GRAPHIC button to open "NIPER GRAPH CONFIGURATION" panel (shown in Fig. 7.2). Inside this panel, use the PLOT CHANNELS ON digital control to select the channels (items) you wish to display. Use arrows to select frame #0 (which corresponds to the channel or item #0). Then look at the adjacent button, which indicates the display status of the selected channel. If the button is in (darker shade), the channel is active for graphical display, otherwise it is inactive. If the button is out (lighter shade), pressing the button once will activate it for graphical display. If the button is in, press the button once and it will return to its lighter shade, indicating that the selected channel is then inactive.

Examine all of the available channels one by one, starting from channel #0. Set the status of each channel to insure that only those channels you want to display are activated. After checking the channels, decide whether or not to use the current settings. Press the ACCEPT button to use the current settings. Press the CANCEL button to disregard any changes and return the settings to their previous values. Both of these buttons will perform their respective functions and then return immediately to the "NIPER GRAPHIC FACILITY" panel. If the new settings are to be saved for future use, follow the instructions for log functions in Chapter 7.

To Export and View Data in External Applications

After the user has selected the set of data to be displayed using other graphic functions described in this section, and likes the display in the "NIPER GRAPHIC FACILITY" plot, the user may want to take a closer look at the data by exporting it to a more advanced graphical application. To accomplish that:

1. Press "External Graphic" button (#4 on Fig. 7.1) located in the top control bar. It will ask you whether you want to return to LabVIEW.
2. Select "YES" if you want to automatically return to LabVIEW after a brief display of data and graphics in the external application. When this option is selected, LabVIEW is activated as soon as display is completed and all the files pertaining to the external application are closed. The display can be either static or dynamic with automatic rotation of the plot to show at various angles, heights and perspectives.

3. Select "NO" if you want to work with the graphics in the external application. When this option is selected, all the related files in the external application stay open after the display is over, and user is free to use full functions of the external application to view, edit, and print the data and plot. LabVIEW continues to run in the background. It is recommended to manually close all non-LabVIEW files and applications when not in use to minimize the memory usage and system failures.

The example driver program "NIPER External Graphic Driver" will open Microsoft Excel application and the Excel Macro Files automatically and display data in graphical form. This example program can be copied and modified for other applications that support "macros" or "scripting." The macros will have to be written by the user using the language provided by the application desired to be interfaced with LabVIEW. The sample macros provided in the "Macro Files" sub-folder are written with adequate comments to help in this task.

To Review Previous Plots

1. Select the run number you want to display using the control box "RUN NO." (#11 in Fig. 7.1). The run number selection has to be between 1 and "NO OF EX-RUNS" (#10 in Fig 7.1). If a selection outside this range is made, it is coerced to the closer of "1" and the last run number.
2. As soon as a new selection of run number is made, the corresponding data is automatically displayed. If it does not for some reason, press "REVIEW EX-PLOT" once.

To Browse Through Previous Plots Sequentially

1. Select the run number with which you want to start browsing by using the control box "RUN NO." (#11 in Fig. 7.1). The run number selection has to be between 1 and "NO OF EX-RUNS" (#10 in Fig 7.1). If a selection outside this range is made, it is coerced to the closer of "1" and the last run number.
2. Press up or down arrow to browse through the graphs sequentially in increasing or decreasing order, respectively. When the selection falls outside the range, it is coerced to the closer of "1" and the last run number.

To Review Plots from an Archived Log File

Method A

1. Open the "NIPER GRAPHIC FACILITY." If it does not automatically start running, run it by clicking on the arrow on the control bar.
2. Press "ADJUST GRAPHIC" (#8 in Fig. 7.1). It will open the "NIPER GRAPHIC CONFIGURATION" panel. Type the full path name of the file in "LOG DATA FILES FULL PATH NAME" box.
3. Press "ACCEPT", which will take you back to "NIPER GRAPHIC FACILITY."

4. Press "STOP" on the "NIPER GRAPHIC FACILITY" panel (#1 in Fig. 7.1).
5. Run the "NIPER GRAPHIC FACILITY" again by clicking on the arrow on the control bar. Examine the "NO OF EX-RUNS" (#10 in Fig 7.1) box. If it is showing 1 or greater, it has been correctly loaded. If not, re-check the file path name again following steps 2-5. Also, make sure the file is a LabVIEW "log" file.
6. Once the log file has been properly loaded, follow the instructions in "To Review Previous Plots" and "To Browse Through Previous Plots Sequentially" sections to display data.

Method B

1. Open and run the "NIPER GRAPHIC FACILITY."
2. Press "ADJUST GRAPHIC" (#8 in Fig. 7.1). It will open the "NIPER GRAPHIC CONFIGURATION" panel. Memorize the full path name of the file shown in the "LOG DATA FILES FULL PATH NAME" box.
3. Stop and close the "NIPER GRAPHIC FACILITY" VI.
4. Rename the archived log-file to the file name memorized and place it into the folder memorized in step 2 above. For example, if it shows "NIPER LabVIEW Programs: GRAPHICS: Logged Data," then rename your file to "Logged Data" and place it into sub-folder "GRAPHICS" of "NIPER LabVIEW Programs" folder.
5. If a dialogue box informs that a file already exists with this name, then temporarily rename the existing file in the destination folder (e.g., "GRAPHICS" in the example above). Adding an extension number to the name is an easy way to rename and remember, then repeat step 4.
6. Open "NIPER GRAPHIC FACILITY." If it is already open, close it and re-open it. This step may sometimes be necessary to access to the new log-file.
7. Run the "NIPER GRAPHIC FACILITY" again by clicking on the arrow on the control bar. Examine the "NO OF EX-RUNS" (#10 in Fig 7.1) box. If it is showing 1 or greater, it has been correctly loaded. If not, re-check the file path name again following steps 2-7. Also, make sure the file is a LabVIEW "log" file.
8. Once the log file has been properly loaded, follow the instructions in "To Review Previous Plots" and "To Browse Through Previous Plots Sequentially" sections to display data.
9. When done, revert the files whose names and/or locations were changed to original names and place them in their original locations.

To Review Recent Plot

Press the "RECENT PLOT" (#6 in Fig. 7.1) button anytime you want to display data from a test in progress. However, there are three conditions to view current test data: (1) A test is in progress, (2) "NIPER MAIN FACILITY" is running, and (3) The "UPDATE GRAPH" on the "NIPER MAIN FACILITY" is pressed. If any of these conditions are not met, and this button is pressed, the "Run Information" box will display the current data, time, and the Run # "Current," but nothing will appear in the graphic palette or the "DATA & DIAGNOSTICS" (#15 in Fig. 7.1) boxes because there is no data available. The "GRAPHIC MESSAGES" (#16 in Fig. 7.1) box will display the message "WARNING: DATA FOR THE FOLLOWING CHANNEL(S) IS NOT AVAILABLE FOR GRAPHIC DISPLAY:" and then lists all the channel numbers selected to display data.

To Curve-Fit Data

To obtain a curve-fit of data, press the "CURVE FIT" (#7 in Fig. 7.1) button. When this button is pressed, any plot the user selects to display on the graphic palette (#18 in Fig. 7.1) will have curve-fit data superimposed on the original data. The type of curve-fit can be selected by the following procedure.

Open the "NIPER GRAPH CONFIGURATION" panel (Fig. 7.2) by pressing the "ADJUST GRAPHIC" (#8 in Fig. 7.1) button. Select the type of curve-fit desired using the box labeled "REGRESSION ANALYSIS." This control box determines the type of curve that is to be fitted to the displayed data. LINEAR, POLYNOMIAL, and EXPONENTIAL are the three possible choices.

If either LINEAR or EXPONENTIAL was selected from "REGRESSION ANALYSIS" box, then you can ignore the "ORDER OF FIT" box. However, if the POLYNOMIAL option was selected, you must also set the degree, or order, of the polynomial fit. Once the selections have been made, decide whether or not to use them. Press the "ACCEPT" button to use the current settings. Press the "CANCEL" button to return them to their previous values. In either case, you will be returned to the "NIPER GRAPHIC FACILITY" panel. If the new settings are to be saved for future use, follow the instructions for log functions in Chapter 7.

It is important to note that pressing the curve-fit button does not automatically show a plot. The user has to select one of the data review options to get a plot with curve-fit . For example, the user will have to follow the procedure described in the section "To Review Recent Plot" in order to see the recent plot along with curve-fits.

Other Graphic Features

The graphic display on the "NIPER GRAPHIC FACILITY" panel contains many features used in the visual presentation of data. The operation of these features (i.e., use of graphics cursors, scaling of the plot, and highlighting of the plot) is described in the following sections.

To Adjust the Time Base of a Plot

Located below the panel's graphics display (#23 in Fig. 7.1) is a small rectangular display box that shows the time base of the plot. To change the time base (or units) on the abscissa of the plot, press the "ADJUST GRAPHIC" (#18 in Fig. 7.1) button. When the "NIPER GRAPH CONFIGURATION" panel is opened, use the "TIME BASE" box to select the desired option. The valid options are: HOURS, MINUTES, and SECONDS; PORE VOLUMES and PICTORIAL are not supported in the current version. Press accept to use the new settings and return to the "NIPER GRAPHIC FACILITY" panel.

To Position Graphics Cursor for Accurate Coordinates

Two graphic cursors are available to provide accurate values of data points. First locate the cursors on the graphic display. Do this by typing arbitrary X and Y coordinate values in the cursor palette (#22 in Fig. 7.1). These values should be within the X- and Y-axis values of the plot for cursor to be visible. Once the graphics cursor has been located, it can be moved to the desired location on the plot.

Move the graphic cursor by any one of three methods. The first method is using the mouse to drag the cursor around the plot. When you move the graphic cursor off the plot, the display will shift itself accordingly in order to keep the cursor visible. The second method is to enter the plot coordinates of the location in which the cursor is to be positioned. This is accomplished by entering the X- and Y-coordinate values in the corresponding columns of the cursor display (#22 in Fig. 7.1). The graphic cursor will then move to the new location. The last method is to use the arrow buttons located directly adjacent to the cursor icons (#23 in Fig. 7.1). Pressing one of these buttons causes the corresponding graphics cursor to move a small incremental distance in the indicated direction.

To Lock the Cursors on the Plots

The two graphic cursors can be used to obtain accurate values of data readings for the activated channels. First, drag one of the graphics cursors and position it directly onto one of the marked points on a plot, indicating the place where a data reading was taken. The cursor will change shape and color when it is aligned with the point. Release the mouse button. The graphics cursor should lock itself onto that data point. You can then read the actual value of that point from the coordinate display for that particular cursor. If you want to see other data points, use the arrow buttons on the cursor palette (#23 in Fig. 7.1) to jump to other points. Once the cursor is locked, it jumps directly from one point to another when the arrows are used. To unlock a graphics cursor, simply drag it off of the data point.

To Shrink or Expand the Graph Horizontally

Place cursor either on the top or bottom of the blue horizontal strip that lies just outside the white graphic area. In order to expand the plots, press the up arrow key on the keyboard, and click the mouse button as many times as you need to expand. To shrink the graph, follow the same procedure as above, but hold down the down arrow key instead of the up arrow key.

To Shrink or Expand the Graph Vertically

To shrink and expand the graph vertically place cursor on either vertical strip just outside the white graphic area. Now follow the same instructions to shrink and expand as stated in the horizontal technique (previous section).

To Shrink or Expand the Graph Proportionally

Place cursor anywhere in the white area and follow the same instructions to scale-down or scale-up as stated in the horizontal technique. This will resize the graph both horizontally and vertically in the same proportion.

To Highlight a Segment of the Plot

Sections of the plot may also be highlighted for distinction. To highlight a segment of a plot, begin by holding down the mouse button on any plot line shown on the NIPER GRAPHIC FACILITY panel. Then drag the cursor (not the graphics cursor) along the plot line until you reach another data point. The enclosed section of the plot line will then be highlighted by a thicker line of the same color. Highlight as many plot segments as you want.

Customizing Graph Controls and Indicators

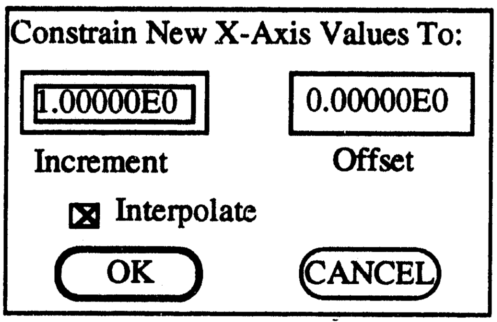
(This section is copied and modified from LabVIEW User's Manual (National Instrument, 1991) and describes how to manipulate a control graph. This information is useful for editing a graph in the front panel of the "Full Size Graph" VI which is opened when the "Edit Graph" button is pressed in "NIPER GRAPHIC FACILITY.)

Changing Plot, Line, and Point Style of Plots

The plot, line, and point styles of graph plots can be changed using the options from the plot pop-up menu or the plot symbol pop-up menu on the graph legend. These pop-up menus appear by pointing to the legend, the cursor, the coordinates of the plotting area, and clicking the mouse while pressing ⌘. The **Plot Style** option pop-ups from the legend displays a palette of plot style choices. For line plots you can use the **Line Style** option to select from several line symbols or choose the **Customize** option to design your own style.

Drawing Graph Data

You can enter plots into graph controls by drawing plots across the plot display window. First, make sure the graph is a control which is the case in "Full Size Graph" VI graph). Then, option-shift-drag the Operating tool across the plot display window. The Operating tool cursor (hand with index finger), normally a cross-hair symbol when it is on the plot display window, changes to a pencil symbol ( ).



By default, X-axis values are constrained to an increment of 1 unit, ensuring an even point density with respect to that axis. This method is useful for drawing waveform plots, although it does not limit you to drawing only single valued functions. To change the default settings choose the **X-axis Constraints...** option from the line plot, point plot, or plot symbol pop-up menu. The dialog box at left appears.

You can change the increment and offset to create an infinite set of possible X-axis values. Remember, an increment of 1 means that a point will be placed one unit from the last point on the X-axis. An offset of zero means a point will be placed at unit values. The default increment of 1 and offset of 0 spaces points at every unit along the axis. But if you want points halfway between unit values, set the offset to 0.5. Setting the increment to 0.5 and the offset to 0 places points at units and halfway between the units. When you draw across any of these values, it is paired with the current Y-axis pixel value and successive pairs are formed into an array of points.

If you set the increment to 0, your drawing is no longer confined to an increment on the X-axis. The increment will be 2 pixels. If you turn off the interpolate option, the increment will vary with the speed with which you draw the plot because the points will be placed at even time intervals. The slower you move the mouse, the higher the point density, and vice versa. When drawing graph data, use **Plots and Lines** mode to see where points are being placed.

If the graph is configured for multiple plots (which is the case in "Full Size Graph" VI graph), they are drawn in order starting from 0. That is, plot 0 must be drawn before you can begin plot 1, and so on.

Deleting Plot Data

To delete a point on the plot, select it with the Operating tool (hand with index finger) and press the Delete key. To delete a plot segment, select the segment by dragging the Operating tool from the first to the last point of the segment to be deleted, then press the delete key. A small superimposed symbol ( ) marks a selected point, and a thick line marks a selected segment.

Editing a Plot

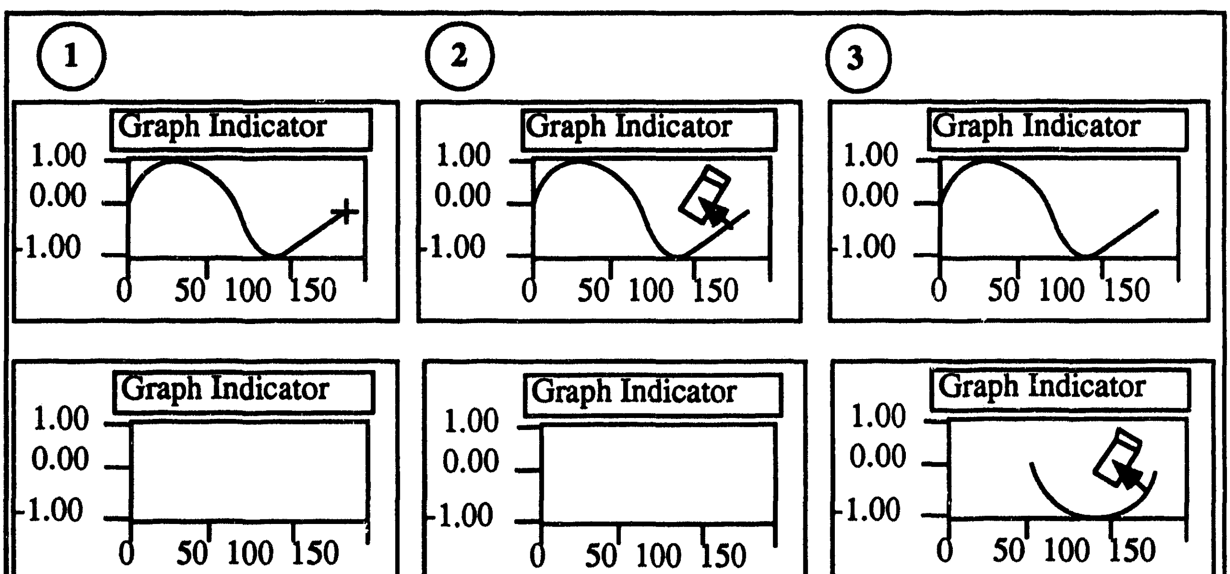
You can extend an existing plot on a graph control, insert new points inside a plot, relocate individual points of the plot, or redraw plot segments to relocate Y-axis values.

To extend an existing plot, simply start drawing (option-shift dragging) from an end point. If X-axis values are constrained, the end point is repositioned at the Y-axis value where the cursor leaves the vicinity of the X-axis value, which does not change. New points are placed at successive X-axis increments. If drawing is not constrained, points are placed at a fixed rate from the moment you begin.

To insert new points inside a plot, start drawing from a position on the plot line that runs between adjacent points. The action breaks the line between those points and inserts new points between them. Should you begin directly on a point, the effect is the same as if you started just to the right of that point. To insert a single point, shift-option-click at the desired position. Inserting is usually done to fill in a plot where the point density is too low.

To relocate any single point, option-drag the point to its new location with the Operating tool.

To redraw plot segments, shift-option-click when you start drawing from a point. Redrawing changes only the Y-axis values of the points. The point from which you begin is repositioned where the cursor leaves the X-axis location. Other points are repositioned where the



Use the Operating tool to select the portion of the plot you want to copy.

With the Datacopy tool, option click on the graph indicator to copy the selected data to the clipboard...

...then click on the graph control to paste the data.

cursor crosses each point's X-axis location. If you redraw too quickly, an intermediate point may not be repositioned. Simply trace back along the new plot route until the point is found and repositioned.

Pasting Plot Data from the Clipboard

You can create or modify a plot by pasting plot data from another front panel control or indicator, even if they are not in the same VI. First, select a plot or part of a plot. Option-click on the plot display window with the Datacopy tool (bottom right on the tool palette). Then click on the target plot display window and the data is copied.

Multiple plots can also be pasted into a control but of course cannot be inserted inside an existing plot. If more plots exist in the Datacopy tool than the control is configured for, the excess plots are ignored. The data in the clipboard must have originated from a graph control or from the equivalent type of numeric control, specifically, a cluster array of points of a cluster array of plots.

Trouble Shooting

SYMPTOM No. 1: LabVIEW quits because of memory shortage or the icon changes to tractor frequently when the program is run.

POSSIBLE REMEDIES: Allocate more memory to LabVIEW by taking these steps:

(1) Quit LabVIEW application. Be sure to save all desired files.

(2) Go to Finder.

(3) Find (⌘+F) and select "LabVIEW" application program.

(4) Select the "Get Info" line from the pull-down "File" menu under the "Finder" application (or, press ⌘+I). This will open a screen showing the suggested size and current size in the memory box.

(5) Increase the current size enough to avoid memory shortage problem for LabVIEW.

There is a trade-off here. Allocating more memory to LabVIEW will decrease the memory available for other applications. Thus, if you want to review your data in other applications such as Microsoft Word™ or Excel™ while LabVIEW is running, you may have to assign optimum memory to each application. Experience will eventually help in solving this problem.

SYMPTOM No. 2: The program does not behave as described in the manual.

POSSIBLE REMEDIES: This is likely to happen when user has taken out-of-sequence actions. There is no harm in doing that, and with a little experience, the user will be able to continue from the point at which the user departed from the instructions. A few simple remedies that may solve the problem, are presented in order of increasing probability of success:

(1) Re-run the program.

(2) Close the program re-open it and run it again.

(3) Quit the LabVIEW application, re-open your program and run it again.

(4) Re-start the computer and start all-over again.

SYMPTOM No. 3: When "External Graphic" button is pressed, a dialog box appears informing the user that there is no data to be plotted, whereas the graphic palette is showing a graph.

POSSIBLE REMEDIES: This happens when the "NIPER GRAPHIC FACILITY" program has been run and stopped once, but the panel has not been closed before re-running the program. The graph shown in the palette does not really exist in the computer's memory—it is only there because the screen has not be updated. In this situation, the easiest solution is to display the run again on the graphic palette, and then press the button.

SYMPTOM No. 4: When browsing the previous plots using the arrows of the "DISPLAY RUN NO" control, the system sometimes crashes.

POSSIBLE REMEDIES: This happens when the arrows are pressed with high frequency (too fast). Every time this button is pressed, a request for update is added in the system queue. Each update involves memory swap (allocation/deallocation), and if done too fast, may overload the system. The easiest remedy is to wait until the current update is complete and then press again. This will also allow time to review data. In case a jump in the run number to be displayed is required, type-in that value instead of conducting a sequential search.

SYMPTOM No. 5: When pressing "RECENT PLOT", the graphic palette and the "DATA & DIAGNOSTICS" box do not show anything.

POSSIBLE REMEDIES: Probably, the "NIPER MAIN FACILITY" is not running and/or, the "GRAPHIC UPDATE" button on its panel is not pressed, it has not been set to acquire any data, or the program has not acquired any set of readings after the "NIPER GRAPHIC FACILITY" was run. First, make sure the "NIPER MAIN FACILITY" is running and that the "GRAPHIC UPDATE" button is activated. Then, press "GET NEW DATA" to acquire current data. If the problem is still not solved, see if controls have been properly set by pressing "ADJUST CONTROL" button on the "NIPER MAIN FACILITY" panel.

REFERENCE

National Instruments Corp., 1991. LabVIEW 2 User's Manual, Part No. 320244-01, Austin, TX, September.

Chapter 9

EXAMPLE PROGRAMS/PROBLEMS

SAMPLE PROBLEMS

NIPER's automation software consists of the three main interactive sections: NIPER MAIN FACILITY, NIPER DISPLAY FACILITY, and NIPER GRAPHIC FACILITY. Each of these facilities are used in the sample problems described in this chapter. Each problem is briefly described, and the operator is asked to configure the software to accommodate the operation of a different experiment or plant set-up.

Problem 1—Operation of an Electronic Balance

Using NIPER's Lab WARDEN software, control an electronic balance (Mettler® PJ-15) using the modem port or the printer port of a low-end Macintosh computer. Using the software, record the weights at pre-determined time intervals and then plot the results as a function of time. Use the schematic of the major components as shown in Fig. 9.1. Use the driver for PJ-15 Mettler balance "DRIVER RS232, Mettler PJ-15," located in "Instrument Drivers" folder.

Guidance for the Problem

Problem 1 was designed for inexperienced users. The step-by-step approach used by some of the students (who are first time users) testing NIPER's software are detailed. The problem is intentionally explained in more detail than may normally be encountered in automation to bridge the gap and assist first time users. Additional guidance can be obtained by working through the "Getting Started Manual" supplied with the Lab VIEW 2 software by National Instrument (1991a) and "Training In-Depth Course on Lab VIEW 2" (1991b). The RS-232 connection for the Mettler PJ-15 electronic balance is shown in detail in Fig. 9.2. Refer to the Mettler PJ-15 user's manual for the description of the electronics within the balance (Mettler, 1991).

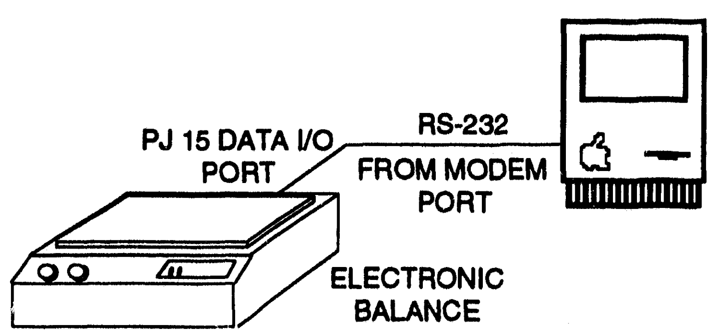


FIGURE 9.1 - Schematic of electronic balance and computer set-up for problem 1.

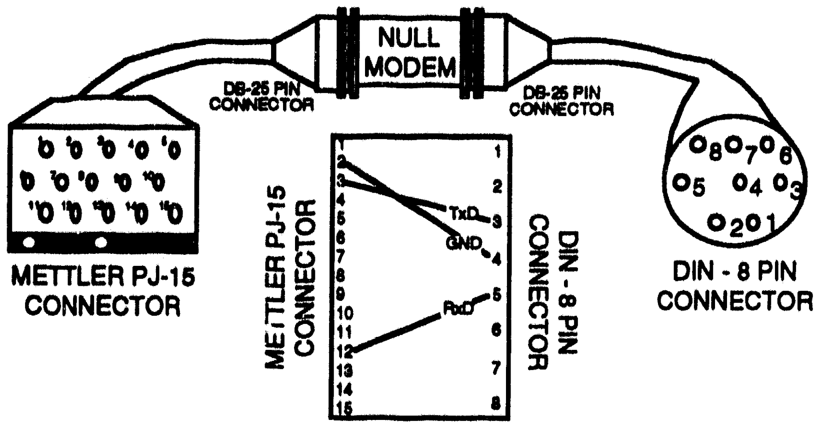


FIGURE 9.2 - Schematic of the Macintosh and Mettler PJ-15 pin configuration.

Sample Solution for Problem 1

1. Install the driver for the Mettler PJ-15, as shown in Fig. 9.3, using the installation procedure under Installing The Driver to the Automation Program section of chapter 4, if it has not been already installed.
2. Using the "Find" command (Fig. 9.4), locate the NIPER Lab WARDEN program (Fig. 9.5), or select it from pull-down apple menu (if you have installed it there). See Installation of NIPER Lab WARDEN section in Chapter 2. Use the program to open and run the "NIPER MAIN FACILITY." It will open "NIPER I/O Facility" panel (Fig. 9.6).
3. After reviewing the NIPER I/O Facility, click the blue button.
4. The NIPER Indicator Panel (Fig. 9.7) now appears. Refer to chapters 3 and 4 of the manual for a thorough description of the NIPER Indicator Panel.
   a. Select the "Frame No." to be 0. Note that the frame number corresponds to the channel number in the NIPER GRAPHIC FACILITY.
   b. Choose a name for the instrument next to the label "Item Name."
   c. Next set the "ITEM NO" to 3. The number 3 was chosen at random. If other instruments are being used make sure they do not have conflicting item numbers.
   d. Below the "ITEM NO" are 8 parameters that pertain to the data values that are read from the balance. Set these ranges as desired.
   e. Now set the parameters under the "Communication Information." Since the balance is hooked to the modem port, the board number is set to 0.
   f. Here the channel number is irrelevant because a board with multiple channels is not being used. Recall that the modem serial port is a unique singular port.
   g. Now choose the correct "Driver VI." Remember that the case number of the location of the driver is now selected in the Driver VI box. The case number of the window in which the PJ-15 driver was installed was 5 (see step 10 in the section mentioned in step 1 above). Therefore, select "DRIVER 5" in the DRIVER VI box.
   h. Under "Common Configuration," set the interval of time between each data reading to 300 seconds. Again, this value was assumed to be desirable for this experiment.
   i. Set the "Test Status" to "NEW: ZERO TIME."
   j. The "MULTIPLEX BOARDS" selector can be ignored since no multiplex boards are being used in this example.
   k. After adjusting the parameters, click outside any control box (but still on the panel).
   l. Select "Make Current Values Default," which is located under the menu option "Operate," Fig. 9.8.
   m. Press the "Accept Current Changes" button when finished. A dialogue box will ask whether to save changes. Respond as you wish.
5. Now the Control Panel (Fig. 9.9) appears; since the balance is not being used as a control instrument in this example (i.e., it does not receive any command, it only sends data), no changes need to be made here except erasing previous entries or disabling them.
   a. Deactivate all the instruments on this control panel to speed up the response of the balance. To deactivate an instrument, enter -1 (a negative integer) in ITEM NO. box.

b. To turn the instruments off, the "SET VALUE" has to be 0 for each frame number that has an instrument active. In this example, there is no active control instrument involved, thus the value in this box is irrelevant, and this step is optional.
c. Select "Make Current Values Default," which is located under the menu option "Operate."
d. Press the "Accept Current Changes" button when finished. A dialogue box will ask whether to save changes. Respond as you desire.

6. Now the NIPER MAIN FACILITY (Fig. 9.10) is ready to take data readings from the balance. The "SUMMARY FOR DIAGNOSTICS" display shows the warning and emergency values (which were set in the NIPER Indicator Panel), recent and previous data readings from the scale, and other parameters. Review "SUMMARY FOR DIAGNOSTICS" display for useful information. For further description and use of the NIPER MAIN FACILITY refer to chapters 3 and 4 of the manual.
   a. Wait for the 10 seconds for data values to be read, or press the button "Get New Data" to obtain additional data readings in addition to the selected scan intervals.
   b. After some data readings have been taken, press the Graph Update button. This will allow data to be graphed on the NIPER GRAPHIC FACILITY.
7. Find and open the NIPER GRAPHIC FACILITY (Fig. 9.11) or use "NIPER Lab WARDEN." Refer to chapters 7 and 8 for detailed information on the NIPER GRAPHIC FACILITY.
   a. The NIPER Graph Configuration Panel (Fig. 9.12) should now be displayed. Under "Data Display Options," select the option "All Data Display."
   b. Select the proper time base and the Y-axis units for the graph to be plotted.
   c. Since we are not curve fitting the graph, ignore the "Order of Fit" and "Type of Curve Fit."
   d. Now select which plot channels need to be turned "ON." Recall that the frame number from the NIPER Indicator Panel (Step 5) corresponds to the plot channel on the NIPER GRAPHIC FACILITY. In step 5, the balance was configured on frame No. 0, thus turn channel #0 ON.
   e. Press the Accept button after making the changes. Respond to the dialogue box as you desire.
8. The front panel now showing is the NIPER GRAPHIC FACILITY. Press the Recent Plot button to display the plot of the data values received from the balance.

Panels and Diagrams for Configuration of Problem 1

The solution to problem 1, configuration of the software for the Mettler PJ-15 balance, is shown in Figs. 9.3 through 9.12 that resulted from sequentially going through the sample solution. The settings shown in these figures are an appropriate set of configurations for the balance. The driver used for this problem to interface with the balance was "DRIVER RS-232 Mettler PJ15" program shown in Fig. 9.13(a-f) (for a legend of this figure, see Vol. 2). This program is included with NIPER Lab WARDEN's library of example instrument drivers. Whereas, configuring NIPER Lab WARDEN does not require much familiarity with LabVIEW (even though it is highly recommended), writing a driver program does require proficiency in LabVIEW.

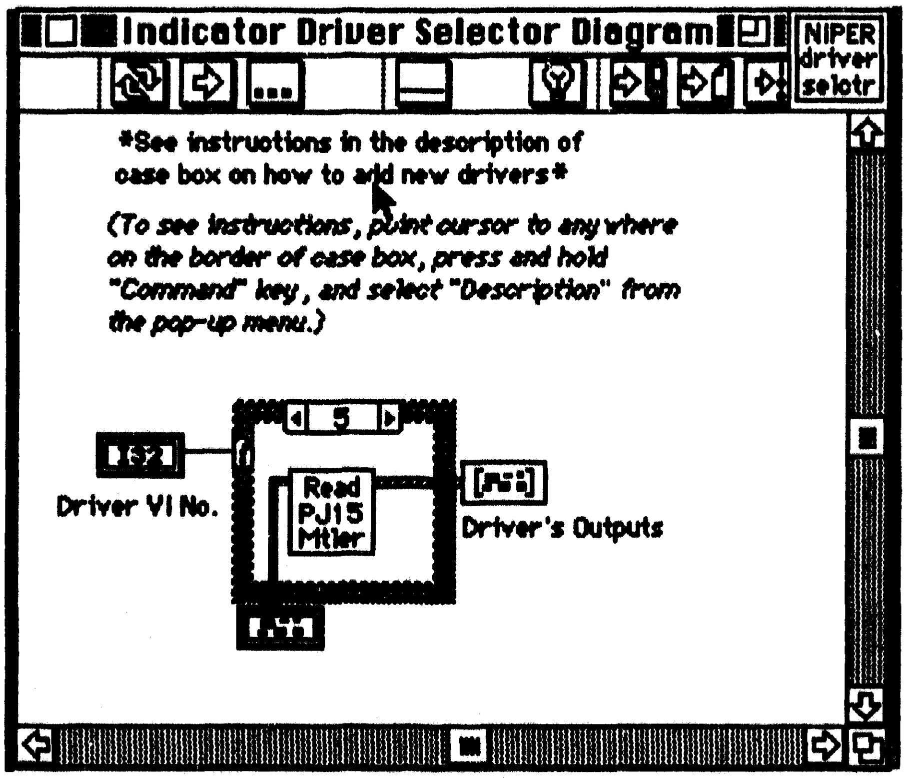


FIGURE 9.3 - Diagram of "Indicator Driver Selector" VI after the driver has been correctly loaded. The icon "Read PJ15 Mtler" was copied from "Instrument Drivers" folder and represents "DRIVER RS 232 Mettler PJ15" VI.

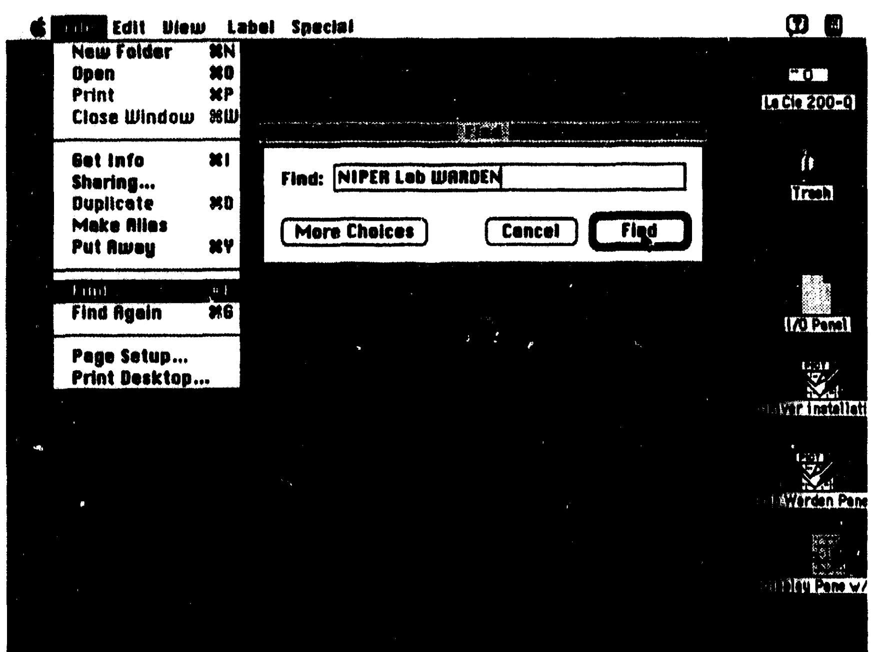


FIGURE 9.4 - Steps involved in using the "Find File" function from the menu.

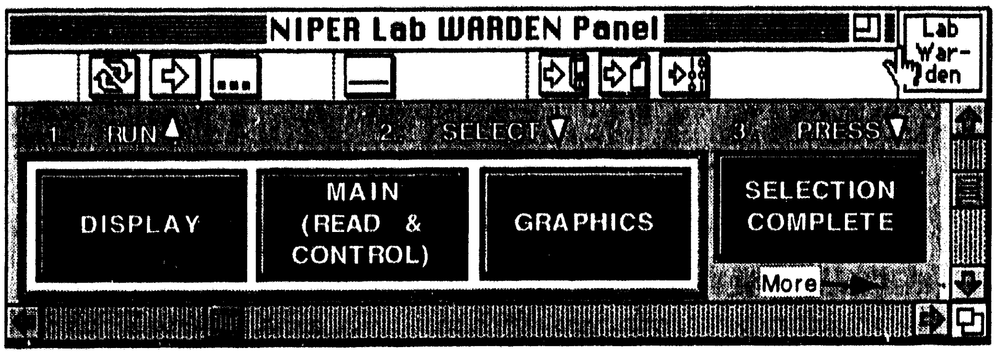


FIGURE 9.5 - Front panel of "NIPER Lab WARDEN" VI. This panel is used to directly access the three main programs.

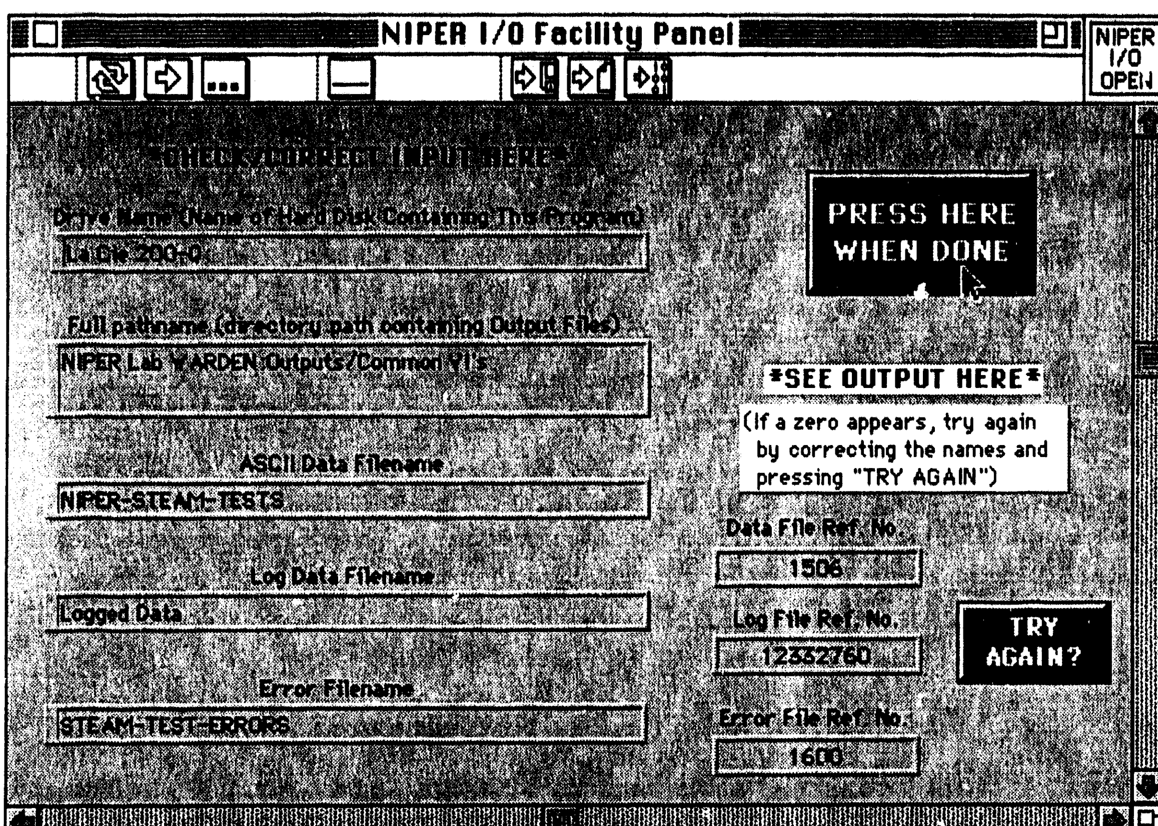


FIGURE 9.6 - Front panel of "NIPER I/O Facility" VI. This panel opens automatically (or manually when operator desires) to allow the selection of filenames and directory locations.

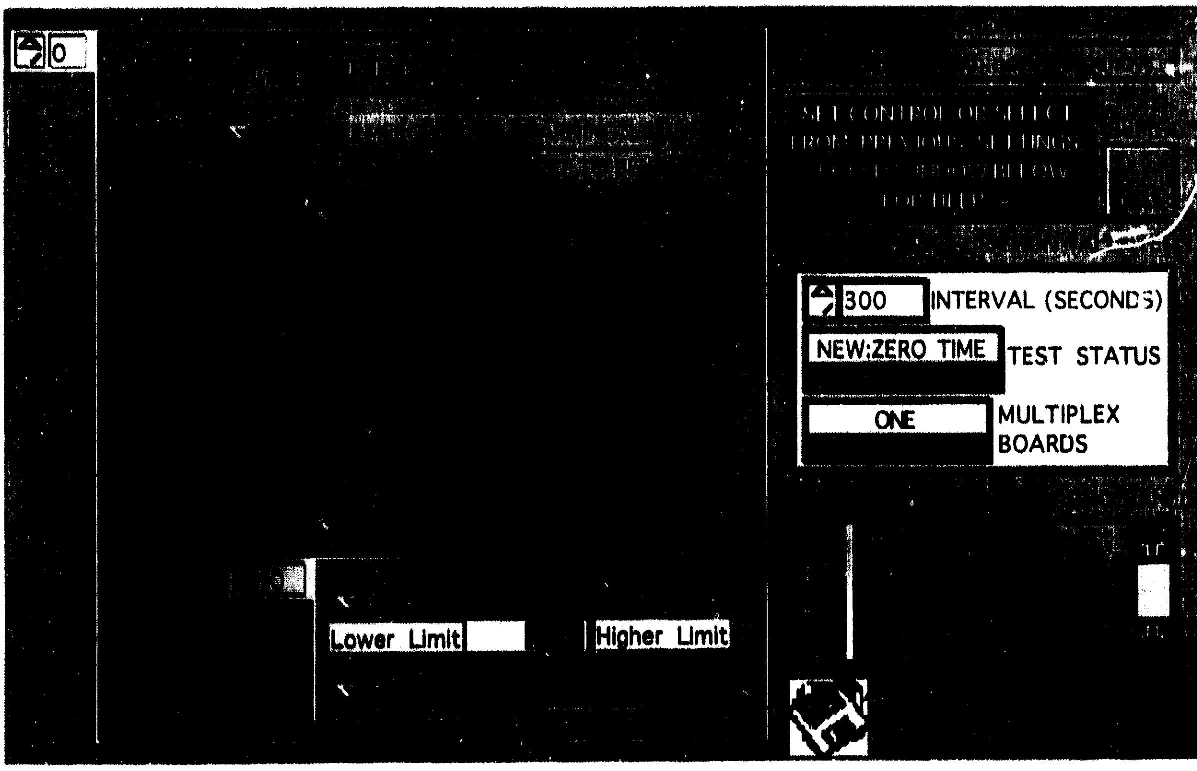


FIGURE 9.7 - Front panel of "NIPER Indicator" VI showing an appropriate set of entries for Problem 1 of this chapter.

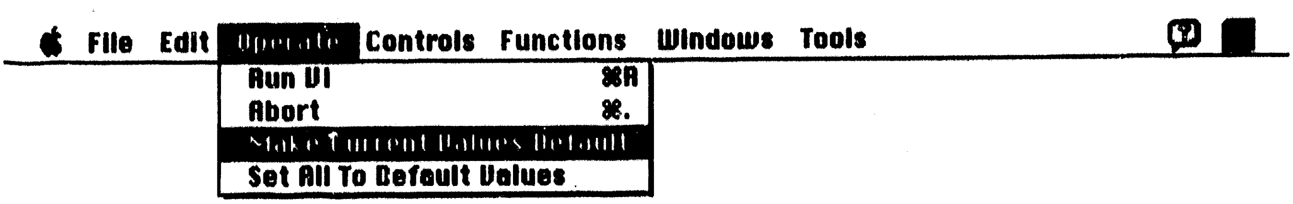


FIGURE 9.8 - How to set default values for the current run from the menu. These default settings will become permanent (till the next selection) if the VI is also saved. Otherwise, the selection will be effective for the current test only.

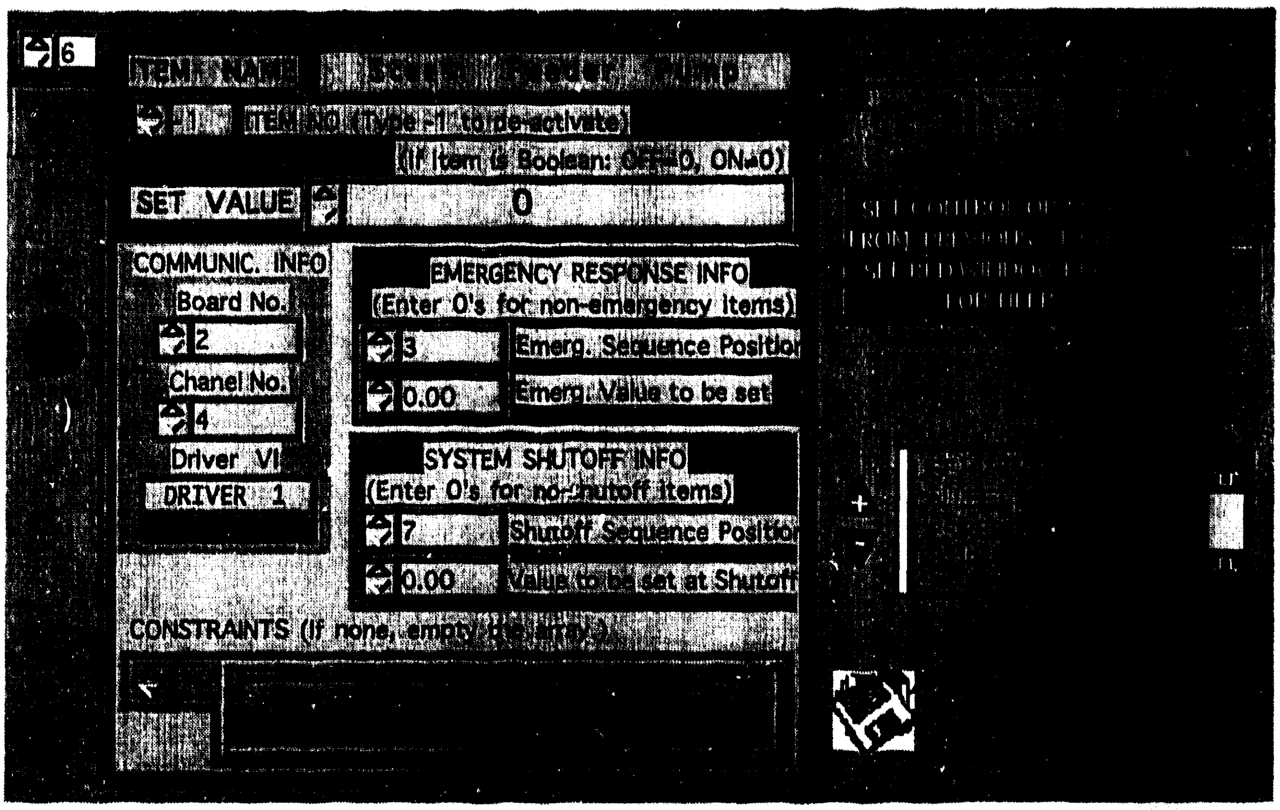


FIGURE 9.9 - Front panel of "NIPER Control" VI showing an appropriate set of entries for Problem 1 of this chapter.

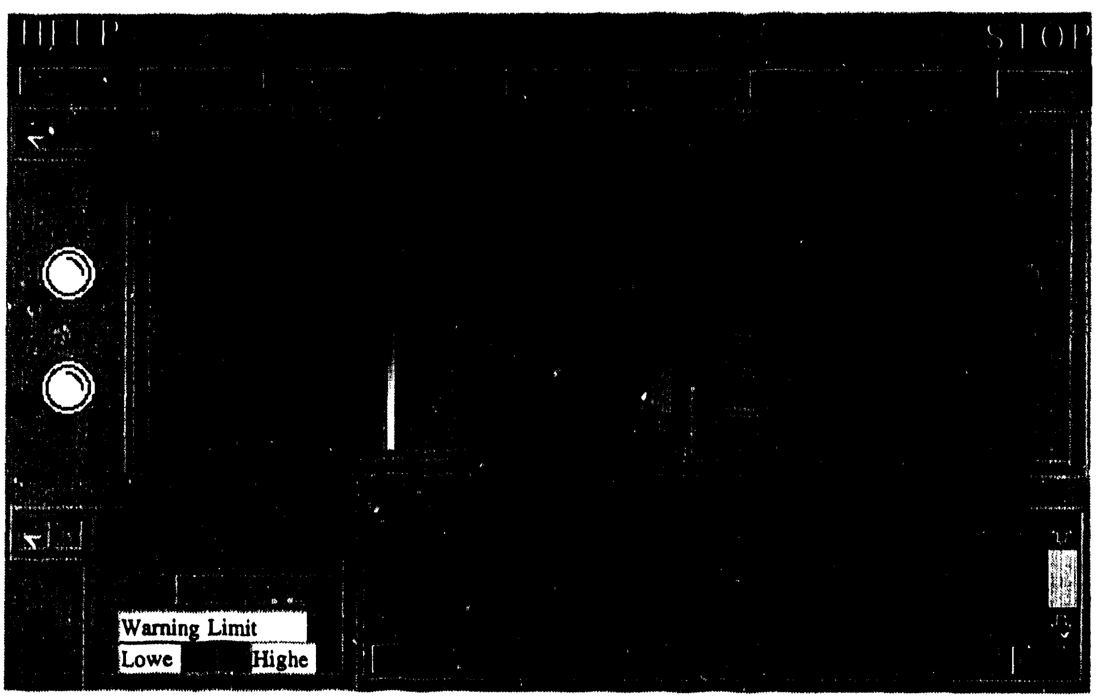


FIGURE 9.10 - Front panel of "NIPER MAIN FACILITY" VI showing the appropriate selection of the menu bar buttons and the type of data received if Problem 1 of this chapter has been properly configured.

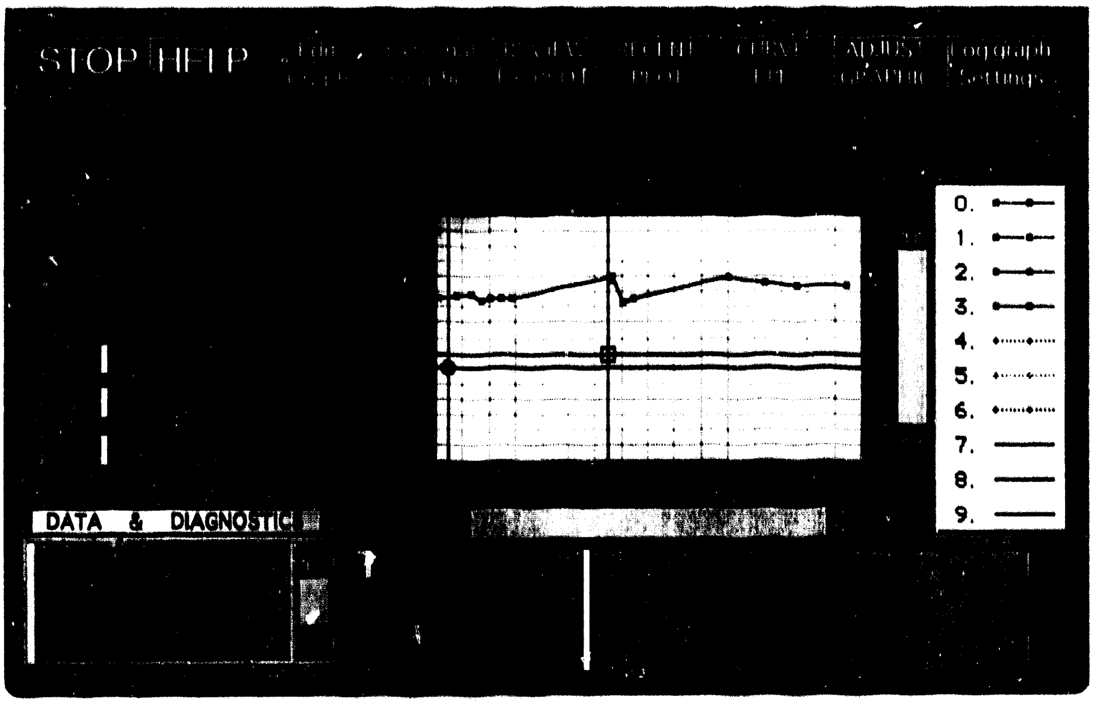


FIGURE 9.11 - Front panel of "NIPER GRAPHIC FACILITY" VI showing the data plot and run information. Note that "RECENT PLOT" button is pressed (not discernible in the figure because of lack of color) to receive data from "NIPER MAIN FACILITY" configured for Problem 1 of this chapter.

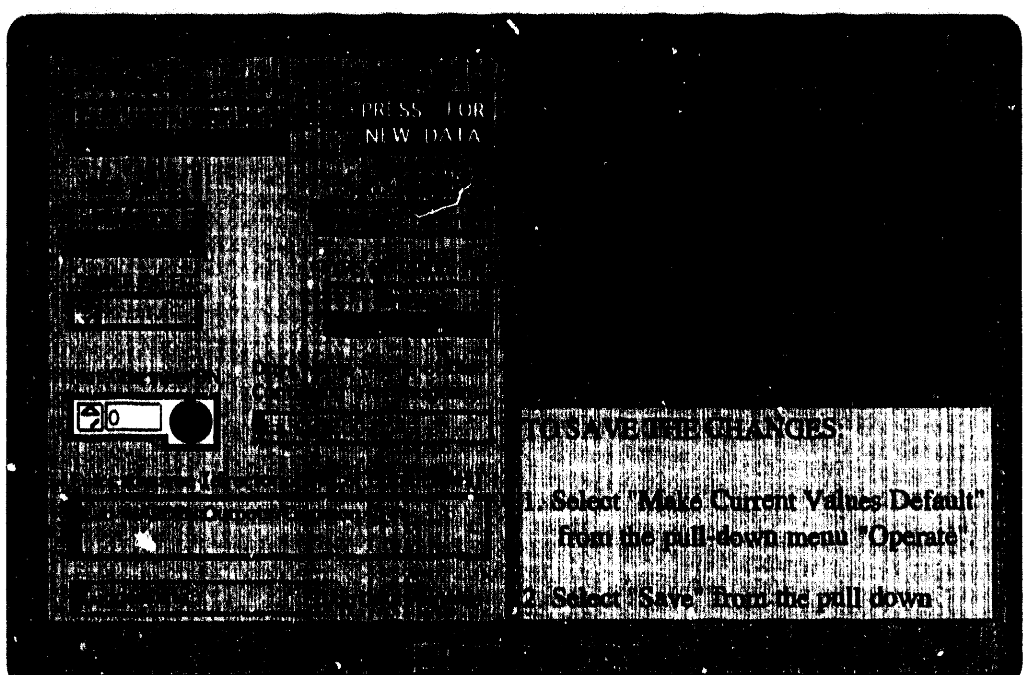
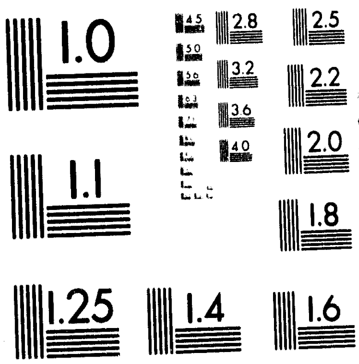
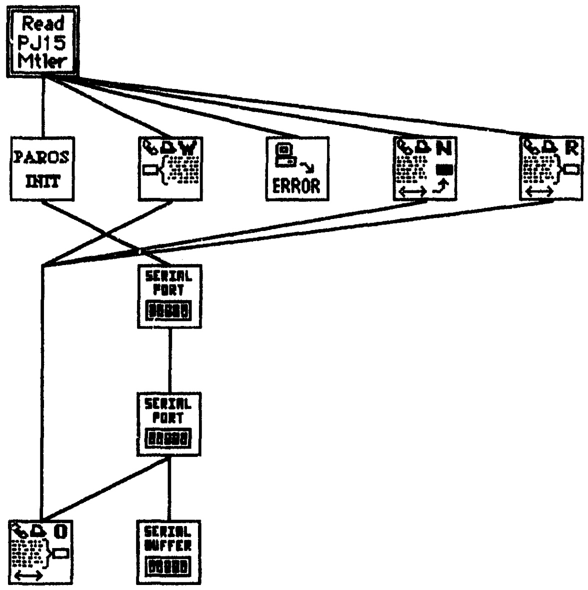


FIGURE 9.12 - Front panel of "NIPER Graph Configuration" VI. This panel opens automatically (or manually when operator desires) to allow the selection of graphic representation of data.

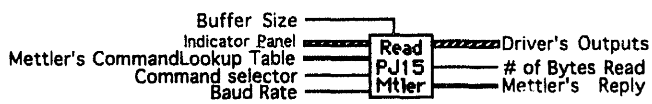

1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

2 of 2

Position in hierarchy



Connector Pane



DRIVER RS232 Mettler PJ15

FIGURE 9.13a - A sample driver program to interface with the balance in Problem 1 of this chapter. The wiring diagrams, when printed in black and white, lose information encoded by wiring the VIs with color. Loss of information also occurs in electronically transferring Lab VIEW screens into Microsoft Word® or similar programs.

Front Panel

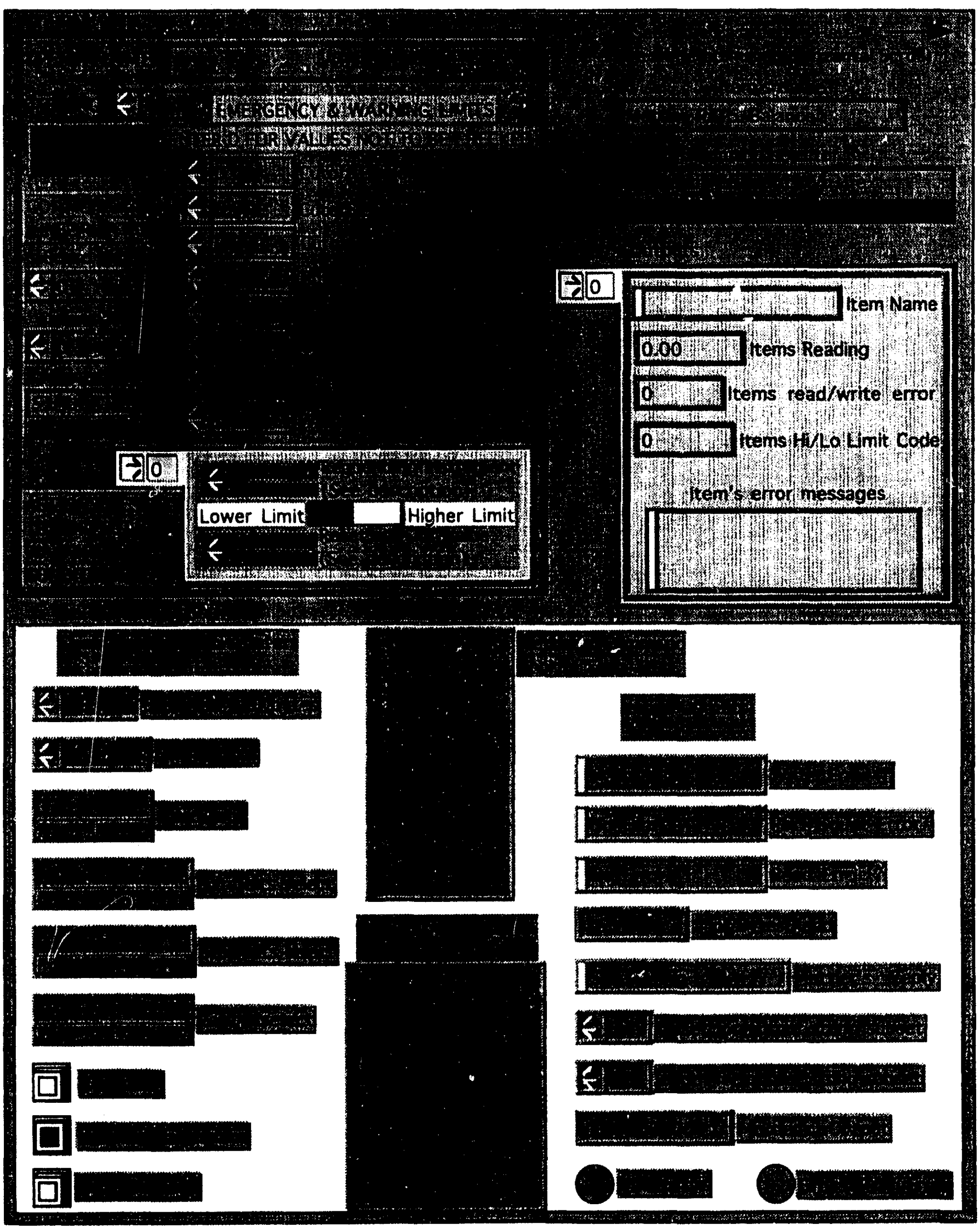


FIGURE 9.13b -A sample driver program to interface with the balance in Problem 1 of this chapter—continued.

Block Diagram

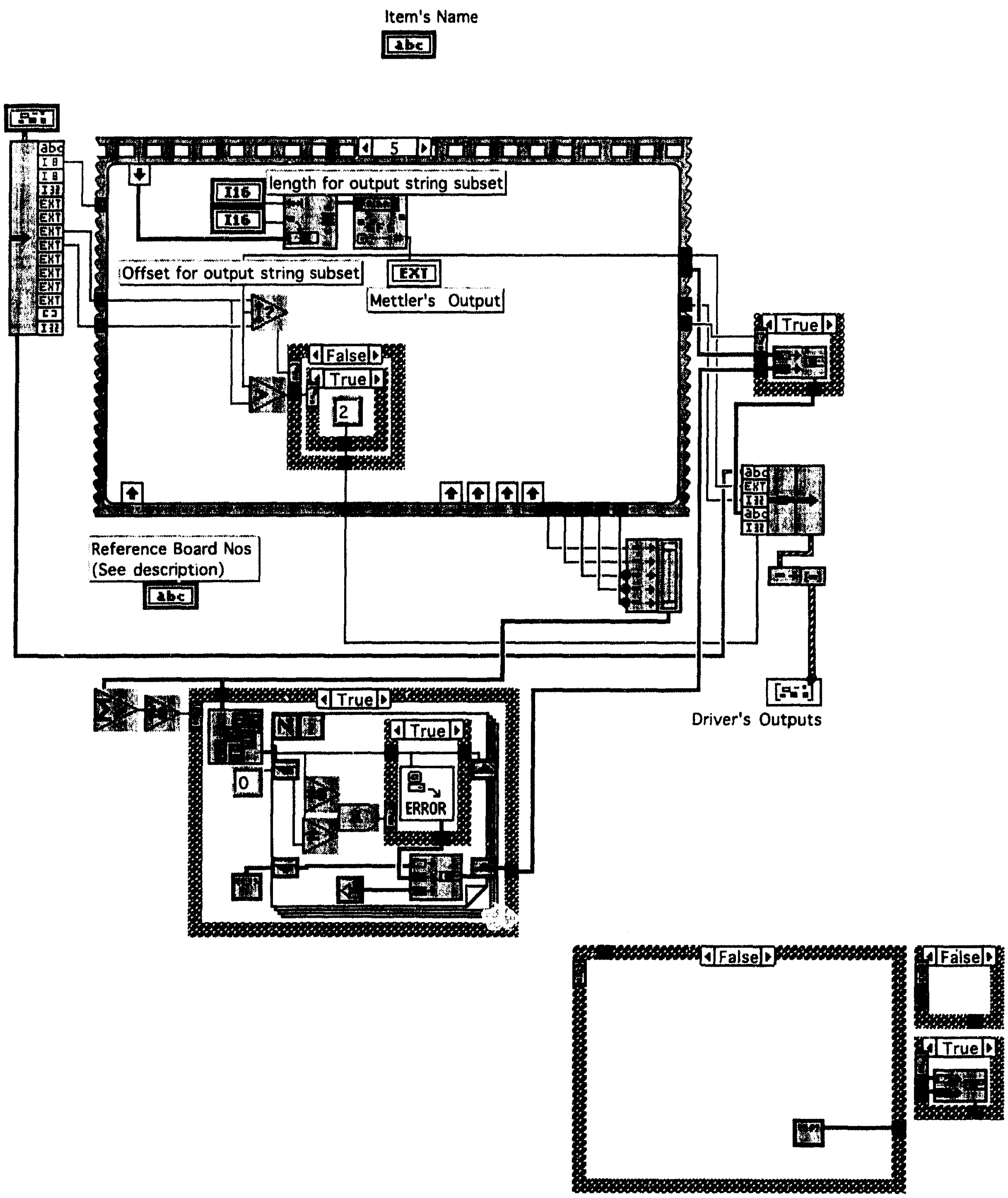


FIGURE 9.13c - A sample driver program to interface with the balance in Problem 1 of this chapter—continued.

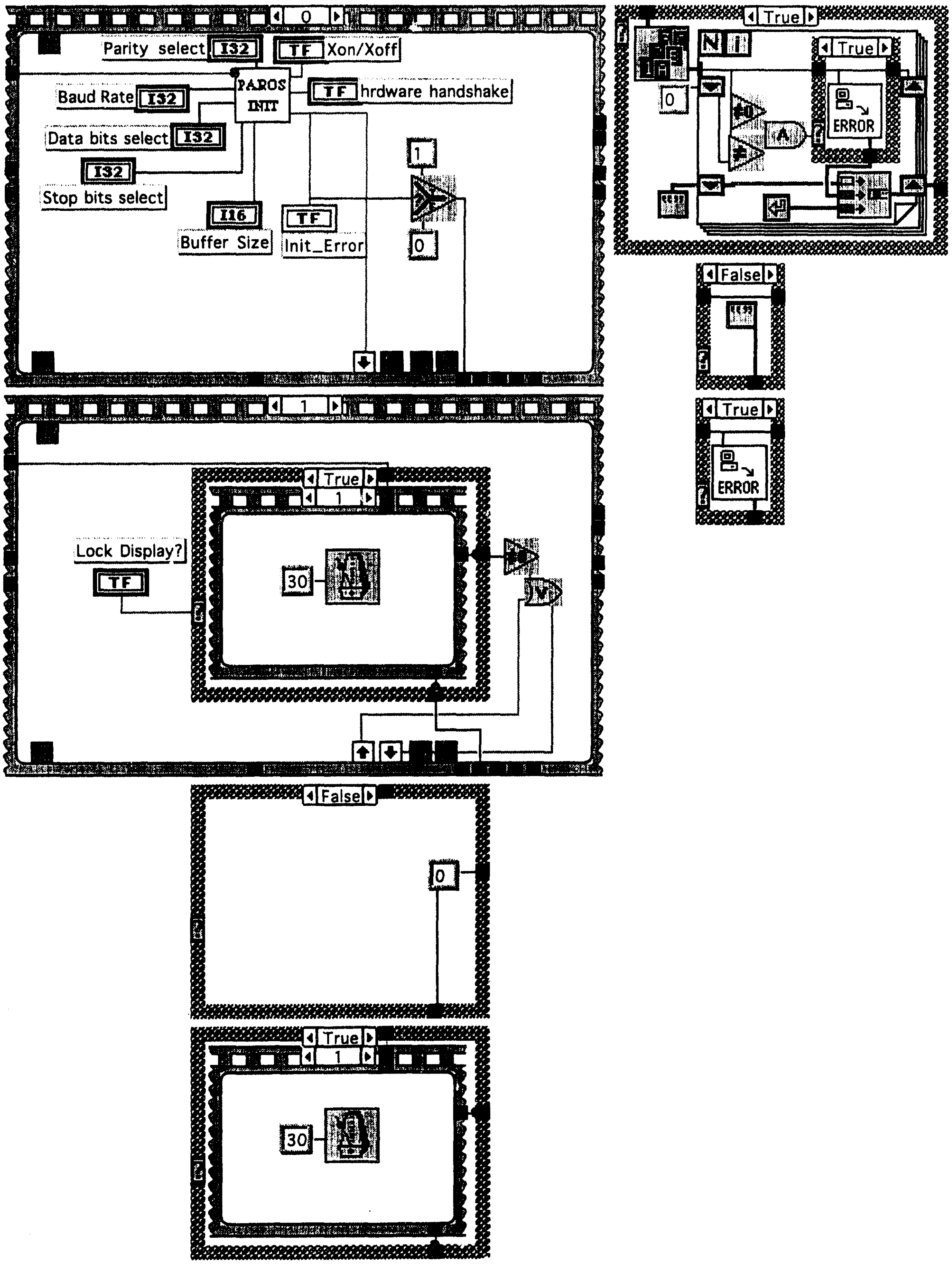


FIGURE 9.13d -A sample driver program to interface with the balance in Problem 1 of this chapter—continued.

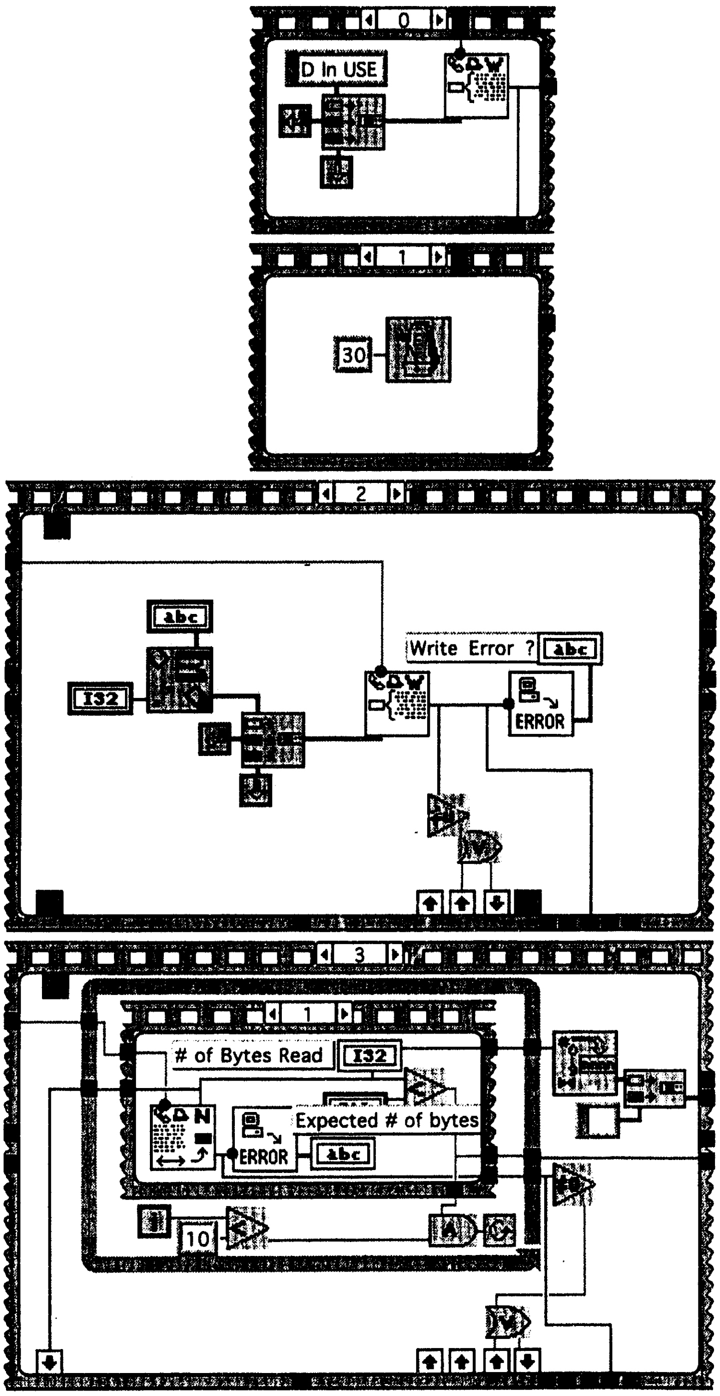


FIGURE 9.13e - A sample driver program to interface with the balance in Problem 1 of this chapter—continued.

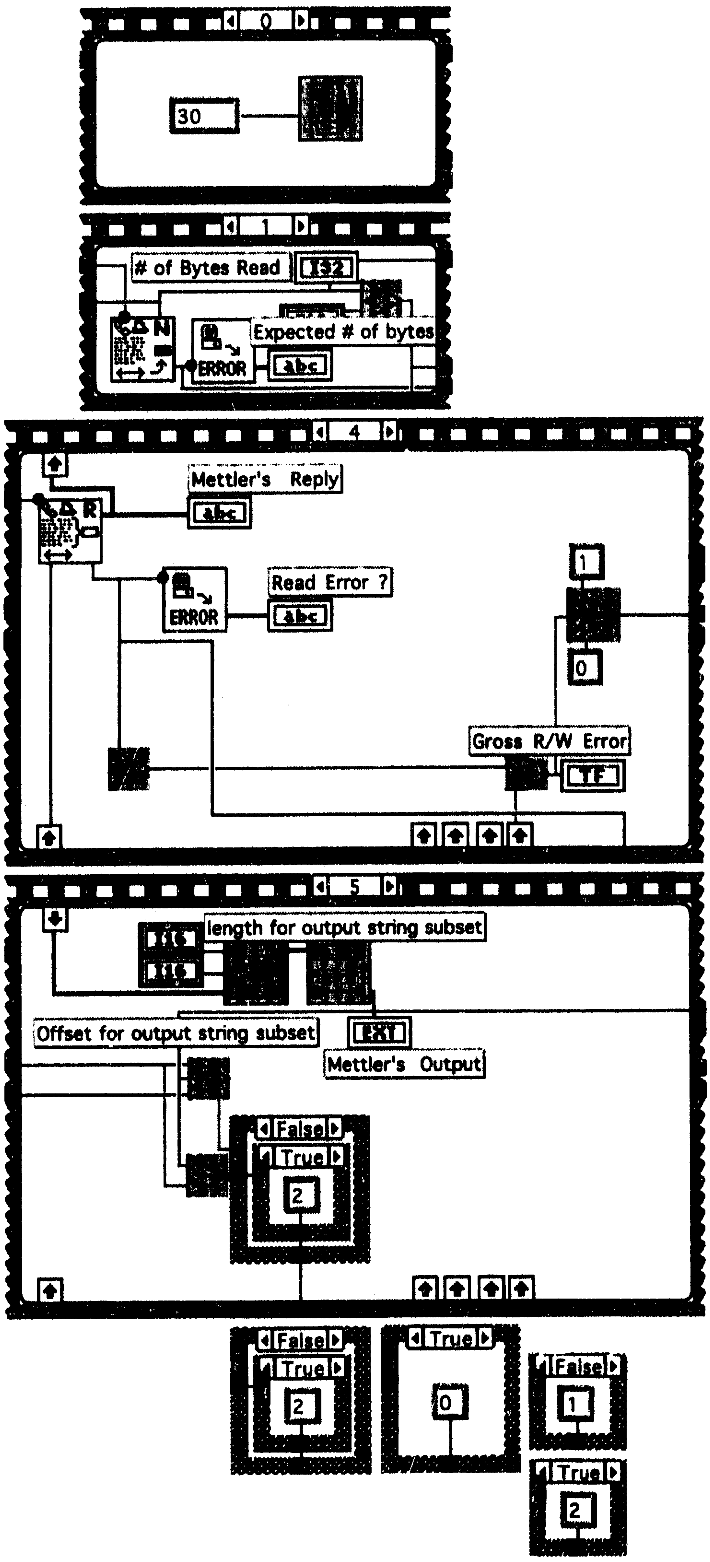


FIGURE 9.13f - A sample driver program to interface with the balance in Problem 1 of this chapter—continued.

Problem 2—Configuration of a Soil Drainage (Permeability) Experiment

A simplified schematic of a soil drainage experiment is shown in Fig. 9.14. The equipment is commonly used in soils and engineering laboratories to determine the rate of water transport (drainage) through soils. Problem 2 is the automation of these permeability measurements. The manual operation of this experiment requires pouring 100 mL of water onto a previously water-saturated column of soil, and manually opening the valve to initiate flow, recording the volume of effluent in a graduated cylinder per unit time. For porous soils, this may only take a few minutes, but for highly compacted soils used to provide a foundation for roadways or buildings, this may require hours. To automate this experiment, the reservoir is filled with 100 mL of water using a computer controlled pump (on/off). The pump rate is manually set to 10 mL/min and the computer turns it off after exactly 10 minutes of pumping. A solenoid switching valve is then opened which allows 100 mL of water to flow through the column. The effluent rate is measured by means of a computer interface balance. The volume is determined from weight assuming the density of water used to flood the soil is 1.0 g/mL.

Sample Solution for Problem 2

Follow the step-by-step procedure described in Chapter 4 section ***To Configure NIPER MAIN FACILITY for Specific Automation Setups.*** The resulting tables from step 1 through 4 are Table 9.1 through 9.4, respectively.

In summary, only two instruments send the signals to computer (the balance and timer), and two instruments receive commands from the computer and change their status (the valve and pump). The problem requires that the valve at start-up should be closed and the pump must

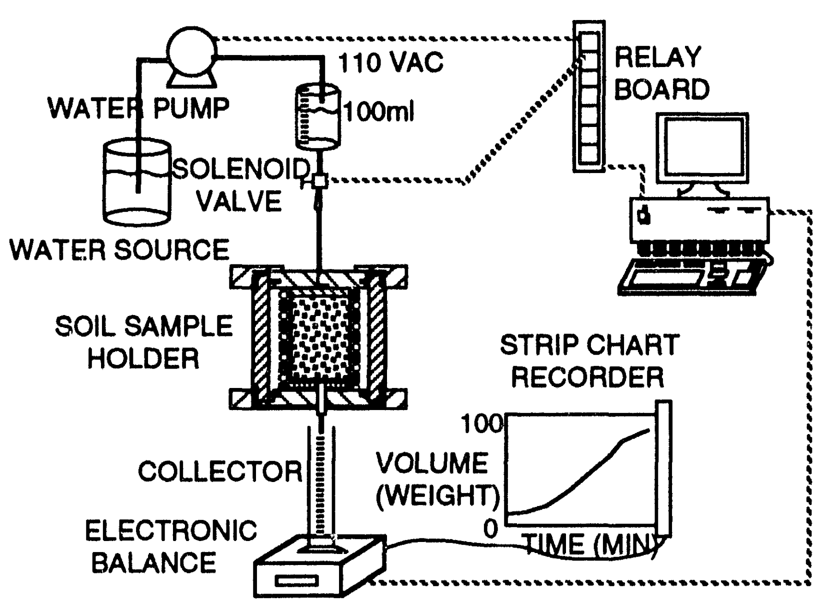


FIGURE 9.14 - Schematic for soil permeability (drainage) apparatus.

TABLE 9.1

Problem Analysis Work Sheet

STEP 1: Enlist All Automation Instruments In Three Categories (See ***To Classify Instruments On The Basis Of Their Functionality*** for more details):

Indicator instruments:	Balance, Timer (computer clock)
Control instruments:	Valve, Pump
Dual-Function instruments:	None

STEP 2: All automation instruments to be reset at start-up (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Valve (OFF), Pump (ON)

STEP 3: All automation instruments to be reset in case of emergency (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): None

STEP 4: All automation instruments to be reset at run shut-off (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Valve (OFF), Pump (OFF)

STEP 5: All automation instruments to be reset in case of warning (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump (OFF), Valve (ON)

WARNING CONDITIONS	CONTROL INSTRUMENTS	RESPONSE
Time ≥10 min.	Valve	Turn ON (Reset to 999)
Time ≥10 min.	Pump	Turn OFF (Reset to 0)
Weight ≥ 90 grams	None	Issue Audio/Video Warning

STEP 6: Enlist All Automation Instruments Whose Operation Is Subjected To Meeting Constraints With Object Instrument(s) (i.e. the control instruments which can be operated only when the status of target instrument(s) meets the constraint conditions):

CONSTRAINED INSTRUMENT	OBJECT INSTRUMENT	LOGICAL CONSTRAINT
Valve	Pump	Valve Can turn ON "ONLY WITHOUT" Pump ON

TABLE 9.2

Settings for Indicator Instruments

BOX NAME	Entry For Set #0	Entry For Set #1
Frame No	0	1
ITEM NAME	PJ-15 Balance	Timer
Item No.	0	1
COMMUNICATION INFORMATION		
Board No.	0	0
Channel No.	0	0
Driver VI	Driver 0	Driver 1
EMERGENCY & WARNING LIMITS		
Cutoff Higher Limit	0	0
Cutoff Lower Limit	0	0
Cutoff Accel. (% per min)	0	0
Cutoff Decel. (% per min)	0	0
Warning Higher Limit	90	10
Warning Lower Limit	0	0
Warning Accel. (% per min)	0	0
Warning Decel. (% per min)	0	0
IDENTIFYING NAME OF THESE SETTINGS	Prob 2 of CH 9	
COMMON CONFIGURATION (ALL ITEMS)		
Interval (Seconds)	60	
Test Status	New:Zero Time	
Multiplex Boards	Select Any	
ACTIONS DESIRED TO CORRECT WARN. COND.		
FRAME NO = 0		
Responding Instrument	Empty	1 (Pump)
Lower or Higher Limit Status	Empty	Higher Limit
Reset at % of Value	Empty	0 (OFF)
FRAME NO = 1		
Responding Instrument	Empty	0 (Valve)
Lower or Higher Limit Status	Empty	Higher Limit
Reset at % of Value	Empty	999 (ON)

TABLE 9.3

Settings for Control Instruments

BOX NAME	Frame 0 Entry	Frame 1 Entry
Frame No	0	1
ITEM NAME	Valve	Pump
Item No.	2	3
Set Value	0	999
COMMUNICATION INFORMATION		
Board No.	2	2
Channel No.	1	0
Driver VI	Driver 0	Driver 0
EMERGENCY RESPONSE INFO		
Emerg. Sequence Position	0	0
Emerg. Value to be set	0	0
SYSTEM SHUTOFF INFO		
Shutoff Sequence Position	1	2
Value to be set at Shutoff	0	0
IDENTIFYING NAME OF THESE SETTINGS	Prob 2 of CH 9	
CONSTRAINTS (If none, empty array)		
FRAME NO = 0		
Logical Condition (No name on box)	Only Without	Only Without
Item on frame number	1	0

TABLE 9.4

List of Drivers and Directory Location(s)

Instrument	Driver Name	Directory Location
Balance	Driver RS232 Mettler PJ15	La Cie 200-Q:NIPER Lab WARDEN: Readout & Control Facility: Instrument Drivers:
Timer	Driver Timer Example	ditto
Valve	Driver Solenoid ON/OFF EXAMPL	ditto
Pump	Driver Solenoid ON/OFF EXAMPL	ditto

be pumping. After 10 minutes of pumping (to have pumped 100 mL @ 10 mL/min), a warning is issued, and as a consequence of this warning, the pump is turned OFF and the valve is turned ON (OPEN). During the test, readings are taken every minute (60 seconds), and the data accumulation starts at the run start-up. As the affluent volume on the balance reaches 90 grams, another warning is issued so that the operator may be alerted that the run termination is at hand. There is no emergency situation expected in this problem. At system shutoff, the valve should be closed. Since it is an unsteady-state test beginning with a constant head, it is necessary that the valve cannot be turned ON while the pump is running.

Hardware Solution for Interfacing With Instruments

Operation of an Electronic Balance

The electronic Mettler PJ-15 balance used in this experiment can be set up as shown in problem 1, but instead of using the modem port, an analog input board (National Instruments) can be used. The RS-232 connection with the balance should still be used. With multiple balances, a 4-port RS-232 board (Greensprings Computer, P/N M51001A) was used. The user should follow the guidance in the manual for the board to configure the relay settings as directed.

Operation of a 110 VAC Solenoid Valve

The soil drainage experiment requires an electronic relay system to operate the solenoid valve. The schematic for a solenoid valve operation is shown in Fig. 9.15 (Circle Seal Controls, 1984). Activation of the valve requires 110 VAC. A single bit digital I/O channel can be used to operate the solenoid indirectly. In order to use such a channel, several interface cards must be used in conjunction with a Macintosh higher series computer. For the digital ports, a National Instruments® NB-DIO24 electronic board was used. The digital signals by themselves are not capable of operating a device such as a solenoid valve; therefore, an additional board with solid state relays must be used. A Gordos® manufactured PB24 relay module mounting board with an OAC5 Module is recommended. These relays are used in place of conventional relays because they are capable of operating directly with the NBDI024 digital I/O board and an additional 5VDC power supply provided by the NB-DIO24 digital I/O board. A schematic of the hardware and relay board is shown in Fig. 9.16.

Operation of a 110 VAC Pump for the Soil Drainage Experiment

The method for operating a 110 VAC pump is exactly the same as the method used to activate the solenoid valve in problem 2. A schematic for the operation of a 110 VAC pump is

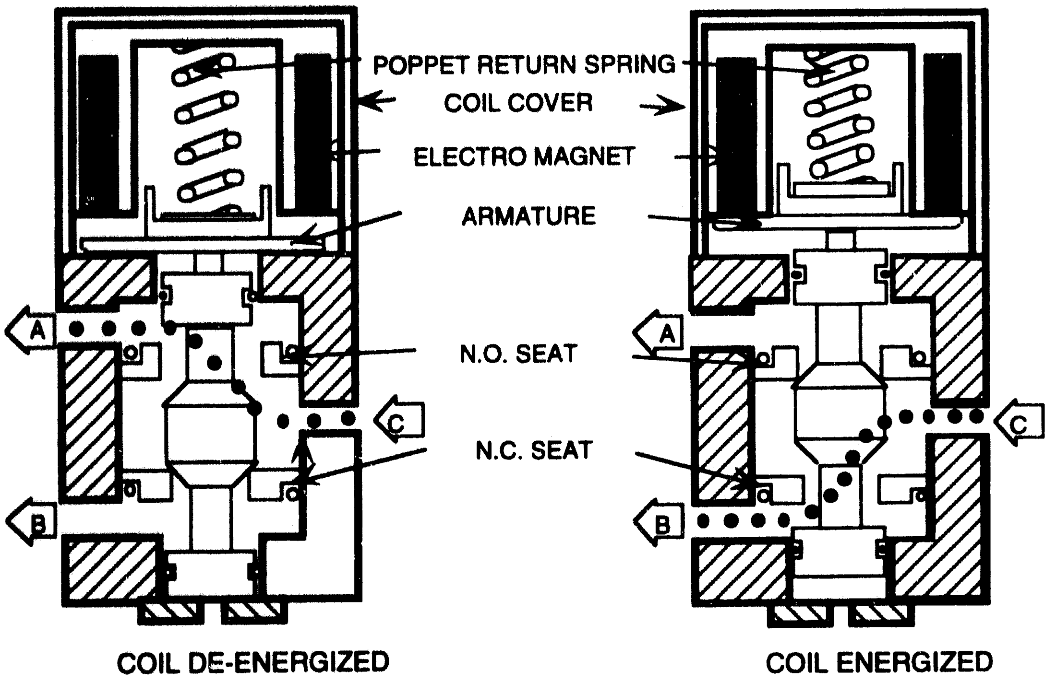


FIGURE 9.15 - Schematic for solenoid valve needed in problem 2.

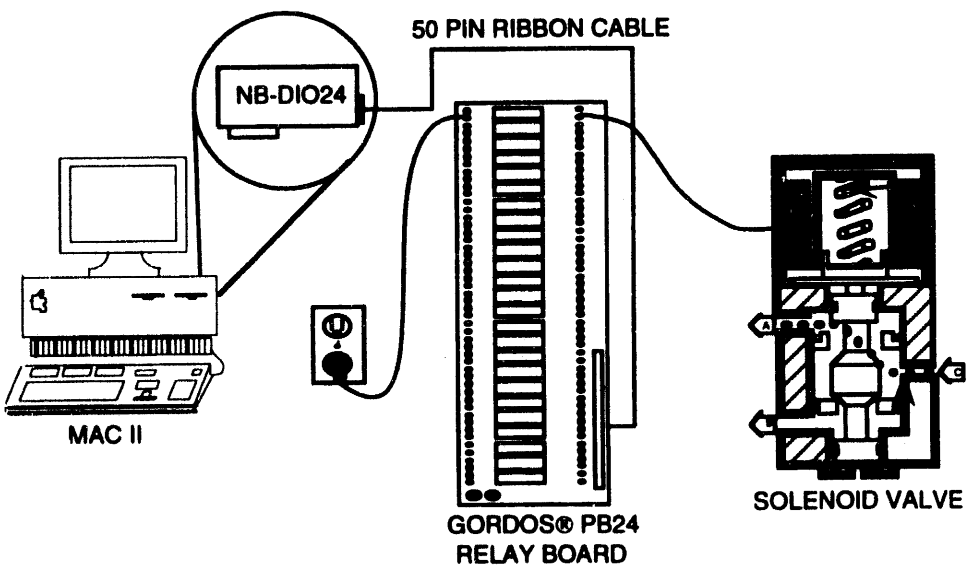


FIGURE 9.16 -Schematic for the solid state relay board for activation of the solenoid valve shown in problem 2.

shown in Fig. 9.17 and the dual relay system to operate pumps or solenoid valves are shown in Fig. 9.18. The NB-DIO24 card has 24 digital I/O channels capable ofbeing used for such a task. Many valves and pumps can be operated using the NB-DIO24 card along with the solid state relay module mounting board and a separate module for each device.

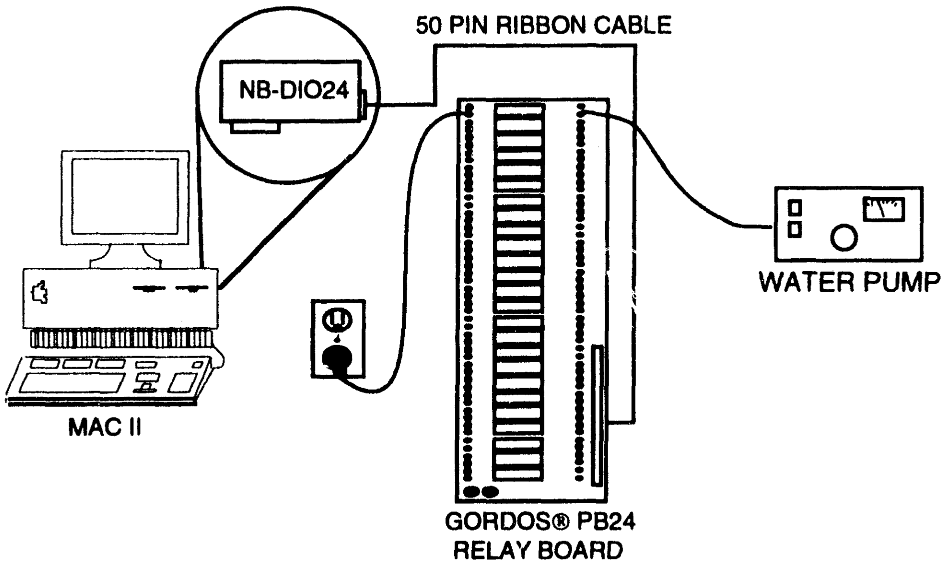


FIGURE 9.17 -Schematic for the operation of a 110 VAC pump for operation of the pump.

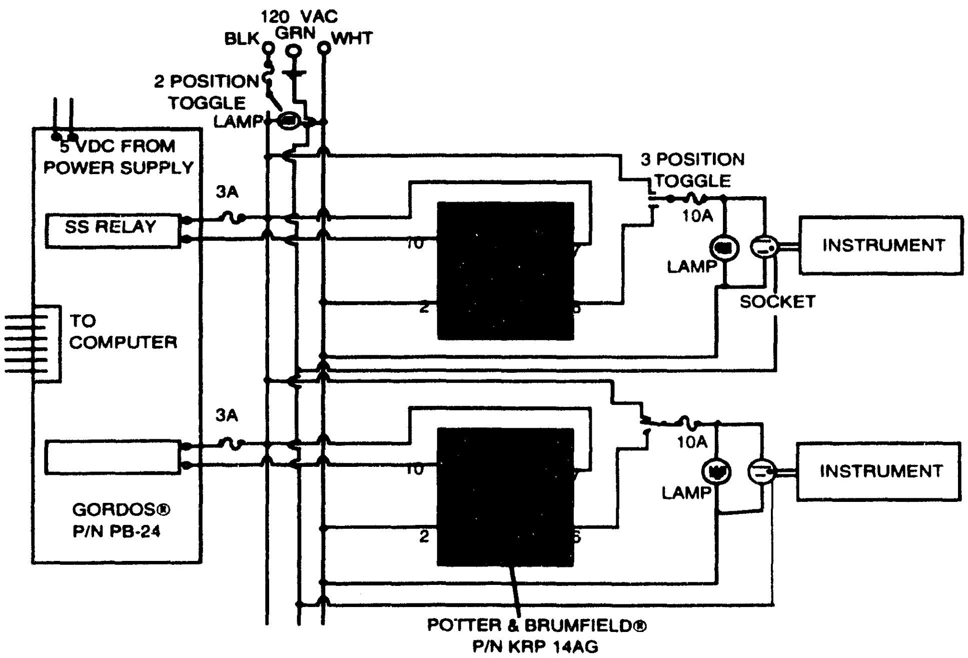


FIGURE 9.18 - Schematic for hardware for dual relay operation of pumps or solenoid valves.

Problem 3—Configuration of a Soil Saturation Experiment

This experimental setup is similar to Problem 2, but it is slightly more complex. The goal of the experiment is to saturate a soil sample, obtain a plot of effluent (volume versus time) to observe how well a soil sample is saturated, and to shut the experiment off automatically when complete saturation is confirmed. A schematic of such a soil saturation experiment is shown in Fig. 9.19. The soil sample is initially unsaturated. The soil sample is flooded under a near constant pressure exerted by the column of water in the beaker that maintains a level of 100±5 mL. The soil sample is assumed to be fully saturated when balance 2 readings on the strip chart almost have a constant slope with respect to time.

As shown in Fig. 9.19, the beaker and collector are both placed on separate balances; both balances are manually tared to zero to remove the weight of the containers. The beaker is initially filled with 100 mL of water. When the experiment begins, the pump is turned off and the valve is opened. The valve stays open throughout the experiment. Whenever the water in the beaker drops below 95 mL (95 grams on balance 1), the pump is turned on. Whenever the water volume of the beaker reaches 105 mL (105 grams on balance 1), the pump is turned off again.

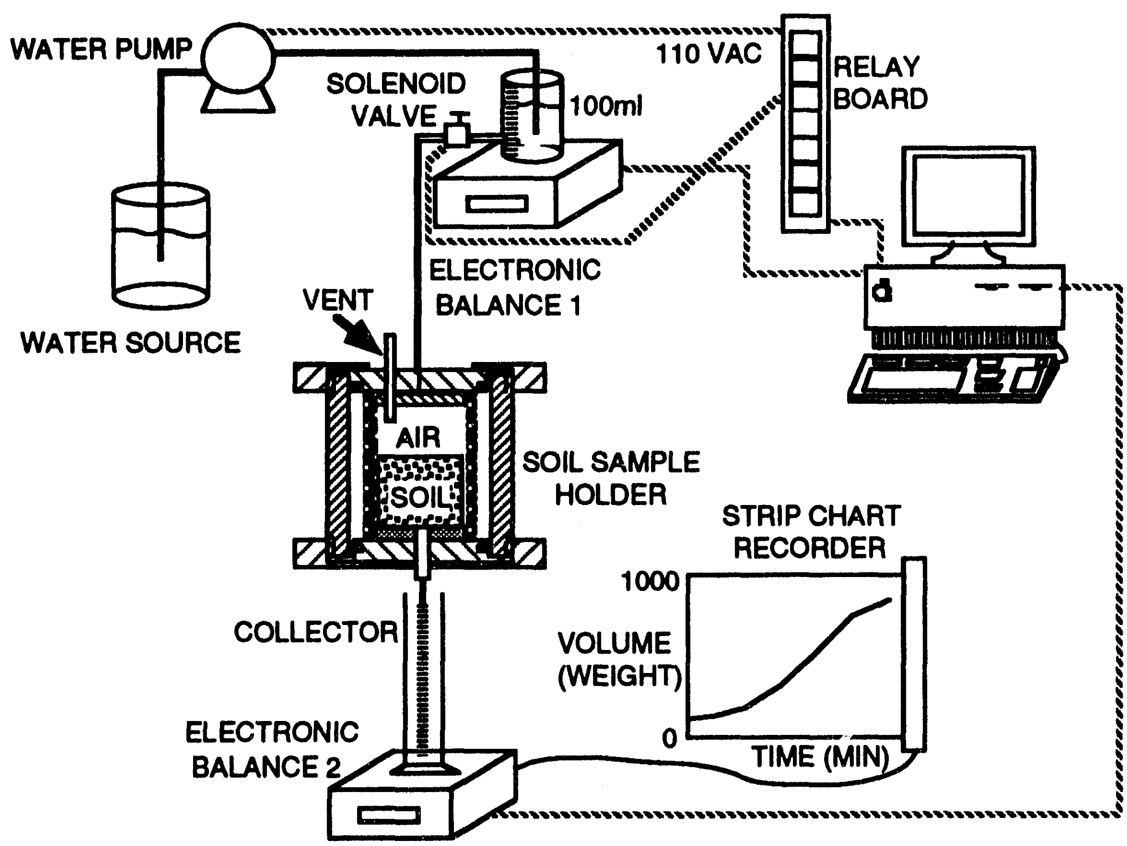


FIGURE 9.19 - Schematic of equipment, Problem 3.

This ON/OFF cycle of the pump maintains a constant level of water in the beaker. When balance 2 readings show a constant rate of volume increase in a one-hour time span, the soil sample is assumed to be saturated; thus, the pump and valve are automatically turned OFF. The first drops of effluent coming from the soil sample would not necessarily imply that the soil sample is fully saturated. That is why a constant slope on the strip chart for balance 2 is used to determine soil saturation.

Some of the operational problems that may arise will require manual adjustments of the pump rate because the configuration (solution for problem 3) will only control the ON/OFF switch of the pump and not the pump rate. The following manual adjustments may be required to maintain a level (100±5 mL) of water in the beaker:

1. If the soil is highly permeable, the pump rate may have to be increased to help maintain the 100 mL water level in the beaker on balance 1.

2. If the permeability of the soil sample is low, the pump rate may have to be decreased in order to avoid overflow of water from the beaker into the sides of the soil sample holder.

Also, it should be noted that this problem does not mandate to turn the valve and pump OFF using the NIPER LAB WARDEN at the end of the experiment because at the end of the experiment the operator may choose to turn the valve and pump OFF anytime using the computer shutoff mechanism. The problem of configuring the shutdown of the valve and pump at the end of the experiment may be pursued as a safety precaution.

Sample Solution for Problem 3

It is highly recommended that Problem 1 and 2 be completed before starting this problem. Follow the step-by-step procedure described in Chapter 4 section ***To Configure NIPER MAIN FACILITY for Specific Automation Setups.*** The resulting tables from step 1 through 4 are Table 9.5 through 9.8, respectively.

In summary, only two instruments send the signals to the computer (balance 1 and balance 2), and two instruments receive commands from the computer and change their status (the valve and pump). The problem requires, at startup, that the valve should be opened and the pump should be OFF. The valve then stays open for the rest of the experiment and is closed only at emergency or normal shut-off. The data accumulation starts at the run start-up to make sure there is no leakage or other problem. During the test, readings are taken every 60 seconds and can be adjusted depending on the permeability. Whenever the weight of water on balance 1 falls below 100 grams, a warning is issued, and as a consequence of that warning, the pump is turned ON. There is a built-in logic in the NIPER Lab WARDEN program which resets an instrument to its start-up value when the warning condition is removed. As the affluent volume on the balance 2 reaches 90 mL (90 grams), another warning is issued so that the operator may be alerted to replace the collector flask. There is no emergency situation expected for this problem, though a hypothetical emergency condition is used to turn the system automatically OFF when the saturation criteria is met (i.e., the rate of volume change in the effluent flask with respect to time become almost constant). At system shutoff, the valve should be closed first, and then the pump be shutoff. To avoid the possibility of spill over, it is necessary that the pump could not be turned ON while the valve is closed.

TABLE 9.5

Problem Analysis Work Sheet

STEP 1: Enlist All Automation Instruments In Three Categories (See ***To Classify Instruments On The Basis Of Their Functionality*** for more details):	
Indicator instruments:	Balance 1, Balance 2, Slope/Curvature
Control instruments:	Valve, Pump
Dual-Function instruments:	None
STEP 2: List all automation instruments to be reset at start-up (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump (OFF), Valve (ON)	
STEP 3: List all automation instruments to be reset in case of emergency (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): None	
STEP 4: List all automation instruments to be reset at run shut-off (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Valve (OFF), Pump (OFF)	
STEP 5: List all automation instruments to be reset in case of warning (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump (ON)	

WARNING CONDITIONS	CONTROL INSTRUMENTS	RESPONSE
Wt. Balance 1 ≤ 100 gms.	Pump	Turn ON (Reset to 999)
Wt. Balance 2 ≥ 90 gms.	None	Audio/Video Warning
Slope/Curvature ≤ 0.2 mL/min^2	None	Audio/Video Warning

STEP 6: List all automation instruments whose operation is subjected to meeting constraints with object instrument(s) (i.e., the control instruments which can be operated only when the status of target instrument(s) meets the constraint conditions):

CONSTRAINED INSTRUMENT	OBJECT INSTRUMENT	LOGICAL CONSTRAINT
Pump	Valve	Pump Can turn ON "ONLY WITH" Valve ON

TABLE 9.6

Settings for Indicator Instruments

Box Name	Entry For Set #0	Entry For Set #1	Entry For Set #2
Frame No	0	1	2
ITEM NAME	Balance 1 (PJ6)	Balance 2 (PJ15)	Slope/Curvature
Item No.	0	1	2
COMMUNICATION INFORMATION			
Board No.	0	1	1
Channel No.	0	0	0
Driver VI	Driver 0	Driver 1	Driver 2
EMERGENCY & WARNING LIMITS			
Cutoff Higher Limit	0	0	0
Cutoff Lower Limit	0	0	0.1
Cutoff Accel. (% per min)	0	0	0
Cutoff Decel. (% per min)	0	0	0
Warning Higher Limit	0	0	0
Warning Lower Limit	100	0	0
Warning Accel. (% per min)	0	0	0
Warning Decel. (% per min)	0	0	0

IDENTIFYING NAME OF THESE SETTINGS	Prob 3 of Chapter 9
COMMON CONFIGURATION (ALL ITEMS)	
Interval (Seconds)	60
Test Status	New:Zero Time
Multiplex Boards	Select Any
ACTIONS DESIRED TO CORRECT WARN. COND.	
FRAME NO = 0	
Responding Instrument	1 (Pump)
Lower or Higher Limit Status	Lower Limit
Reset at % of Value	999

TABLE 9.7

Settings for Control Instruments

Box Name	**Frame 0 Entry**	**Frame 1 Entry**
Frame No	0	1
ITEM NAME	Valve	Pump
Item No.	3	4
Set Value	999	0
COMMUNICATION INFORMATION		
Board No.	2	2
Channel No.	1	0
Driver VI	Driver 0	Driver 0
EMERGENCY RESPONSE INFO		
Emerg. Sequence Position	1	2
Emerg. Value to be set	0	0
SYSTEM SHUTOFF INFO		
Shutoff Sequence Position	1	2
Value to be set at Shutoff	0	0
IDENTIFYING NAME OF THESE SETTINGS	Prob 3 of Chapter 9	
CONSTRAINTS (If none, empty array)		
FRAME NO = 0		
Logical Condition (No name on box)	Empty	Only With
Item on frame number	Empty	3

TABLE 9.8

List of Drivers and Directory Location(s)

INSTRUMENT	**DRIVER NAME**	**DIRECTORY LOCATION**
Balance 1	Driver RS232 Mettler PJ6	La Cie 200-Q:NIPER Lab WARDEN: Readout & Control Facility: Instrument Drivers:
Slope/Curvature	Driver Slope/Curvature Finder	ditto
Balance 2	Driver RS232 Mettler PJ15	ditto
Valve	Driver Solenoid ON/OFF EXAMPL	ditto
Pump	Driver Solenoid ON/OFF EXAMPL	ditto

Problem 4—Configuration of a Coreflood Experiment

Problem 4 is a coreflood experimental set-up to determine the permeability of a rock. This is accomplished by measuring the fluid flow rate and the differential pressure across the core and calculating the permeability from Darcy's Law. The apparatus shown in Fig. 9.20 needs to be automated with the following attributes. The lower warning limit of balance 1 is 100 mL (100 g). If the reservoir level falls below 100 g the reservoir balance warning light will turn ON to warn the user. The lower emergency limit is when the water level goes below 10 g. In this case, the valve and the pump are to be turned OFF.

Balance 2 must be within 100-115 mL (100-115 g). The pump is to be turned ON when the water level drops below 100 g. The pump is turned OFF, and the inlet balance warning light is turned ON when the water level exceeds 115 g. If by mishap the water level exceeds 120 g, the valve will switch to drain, and the experiment will be shut-off because its validity has become

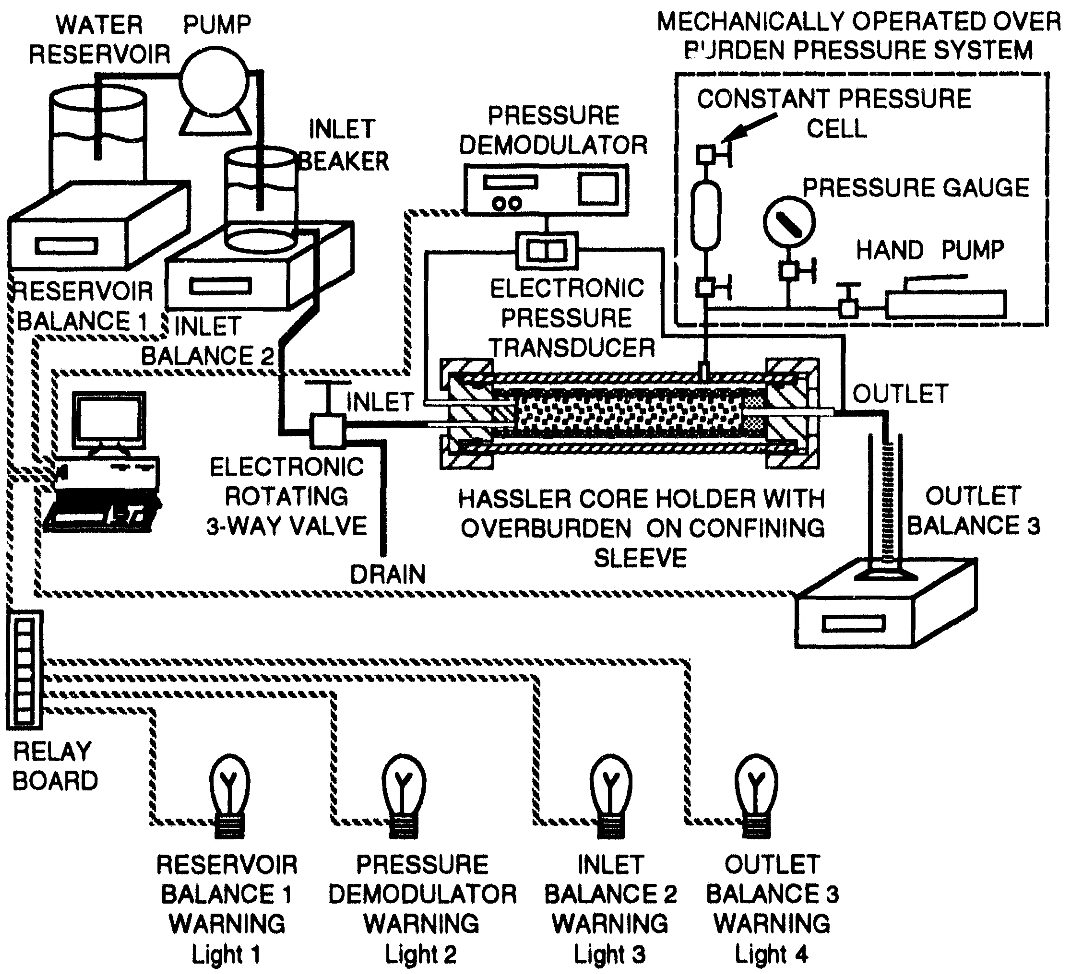


FIGURE 9.20 - Coreflood experimental set-up.

doubtful. The reading of the pressure demodulator from the pressure transducer will range from 0 to 25 psi. If the pressure exceeds 25 psi, the pressure warning light is triggered. If the pressure exceeds 30 psi, the pump and the valve are turned OFF. If balance 3 exceeds 500 grams, the outlet balance warning light turns ON to signal that the outlet beaker is about to overflow.

Sample Solution for Problem 4

It is highly recommended that Problems 1 through 3 be completed before starting this problem. Follow the step-by-step procedures described in Chapter 4 ***To Configure NIPER MAIN FACILITY for Specific Automation Setups.*** The resulting tables from steps 1 through 4 are Tables 9.9 through 9.12, respectively.

In summary, only four instruments send signals to the computer (balance 1 through balance 3, and the pressure demodulator), and six instruments receive commands from the computer and change their status (the valve, pump, and lights 1 through 4). The problem requires, at startup, that the valve should be opened, but the pump and the lights should be OFF. The valve stays open except for draining in case of emergency or at normal shut-off. The data accumulation from all indicator instruments (balances 1 through 3, and pressure demodulator) starts at the run start-up to detect any problems. During the test, readings are taken every minute (60 seconds).

Whenever the weight of water on balance 1 (reservoir balance) falls below 100 grams, the warning Light 1 is turned ON to alert the operator that it is time to add more water into the beaker. Whenever the weight of water on balance 2 (inlet balance) falls below 100 grams, the warning Light 2 is turned ON to alert the operator, and the pump is turned ON. There is a built-in logic in the NIPER Lab WARDEN program which resets an instrument to its start-up value when the warning condition is removed. If the weight of water on balance 2 (inlet balance) exceeds 115 grams, the warning Light 2 is turned ON to alert the operator. Similarly, if the weight of water on balance 3 (outlet balance) exceeds 450 grams, the warning Light 3 is turned ON to alert the operator that the beaker needs to be emptied soon. Finally, if pressure on the system exceeds 25 psi, warning Light 4 is turned ON to indicate possible problems ahead.

In the emergency condition of balance 2 (reservoir balance) reading more than 120 grams, the valve is set to drain (set value = 1), and then the pump is turned OFF. When the operator wants to terminate the experiment, the valve, pump and lights 1 through light 4 are turned OFF in the sequence they are listed. The only constraint in the experiment is that the pump cannot be turned ON as long as the valve is OFF to avoid accidental spill over.

TABLE 9.9

Problem Analysis Work Sheet

STEP 1: List all automation instruments in three categories (See ***To Classify Instruments On The Basis Of Their Functionality*** for more details*):*

Indicator instruments:	Balance 1, Balance 2, Balance 3, Press. Transd.
Control instruments:	Valve, Pump, Light 1, Light 2, Light 3, Light 4
Dual-Function instruments:	None

STEP 2: List all automation instruments to be reset at start-up (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump (OFF), Valve (ON), Light 1 (OFF), Light 2 (OFF), Light 3 (OFF), Light 4 (OFF)

STEP 3: List all automation instruments to be reset in case of emergency (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump (OFF), Valve (DRAIN)

STEP 4: List all automation instruments to be reset at run shut-off (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Valve (OFF), Pump (OFF)

STEP 5: List all automation instruments to be reset in case of warning (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump (ON), Light 1 (ON), Light 2 (ON), Light 3 (ON), Light 4 (ON)

WARNING CONDITIONS	CONTROL INSTRUMENTS	RESPONSE
Wt. Balance 1 ≤ 100 gms.	Light 1	Turn ON (Reset to 999)
Wt. Balance 2 ≤ 100 gms.	Light 2 Pump	Turn ON (Reset to 999) Turn ON (Reset to 999)
Wt. Balance 2 ≥ 115 gms.	Light 2	Turn ON (Reset to 999)
Wt. Balance 3 ≥ 450 gms.	Light 3	Turn ON (Reset to 999)
Press. Transd. ≥ 25 psi	Light 4	Turn ON (Reset to 999)

STEP 6: List all automation instruments whose operation is subjected to meeting constraints with object instrument(s) (i.e., the control instruments which can be operated only when the status of target instrument(s) meets the constraint conditions):

CONSTRAINED INSTRUMENT	OBJECT INSTRUMENT	LOGICAL CONSTRAINT
Pump	Valve	Pump Can turn ON "ONLY WITH" Valve ON

TABLE 9.10

Settings for Indicator Instruments

Box Name	Entry For Set #0	Entry For Set #1	Entry For Set #2	Entry For Set #3
Frame No	0	1	2	3
ITEM NAME	Balance 1 (PJ15)	Balance 2 (PJ6)	Balance 3 (PJ15)	Press. Transd.
Item No.	0	1	2	3
COMMUNICATION INFORMATION				
Board No.	0	2	2	2
Channel No.	0	0	1	3
Driver VI	Driver 0	Driver 1	Driver 0	Driver 2
EMERGENCY & WARNING LIMITS				
Cutoff Higher Limit	0	120	0	30
Cutoff Lower Limit	0	0	0	0
Cutoff Accel. (% per min)	0	0	0	0
Cutoff Decel. (% per min)	0	0	0	0
Warning Higher Limit	0	115	450	25
Warning Lower Limit	100	100	0	0
Warning Accel. (% per min)	0	0	0	0
Warning Decel. (% per min)	0	0	0	0

IDENTIFYING NAME OF THESE SETTINGS	Prob 4 of Chapter 9
COMMON CONFIGURATION (ALL ITEMS)	
Interval (Seconds)	60
Test Status	New:Zero Time
Multiplex Boards	Select Any

ACTIONS DESIRED TO CORRECT WARN. COND.

FRAME NO = 0				
Responding Instrument	2 (Light 1)	3 (Light 2)	4 (Light 3)	5 (Light 4)
Lower or Higher Limit Status	Lower Limit	Lower Limit	Higher Limit	Higher Limit
Reset at % of Value	999	999	999	999
FRAM NO = 1				
Responding Instrument	Empty	1 (Pump)	Empty	Empty
Lower or Higher Limit Status	Empty	Lower Limit	Empty	Empty
Reset at % of Value	Empty	999	Empty	Empty
Frame No = 2				
Responding Instrument	Empty	3 (Light 2)	Empty	Empty
Lower or Higher Limit Status	Empty	Higher Limit	Empty	Empty
Reset at % of Value	Empty	999	Empty	Empty

TABLE 9.11

Settings for Control Instruments

Box Name	Frame 0 Entry	Frame 1 Entry	Frame 2 Entry	Frame 3 Entry	Frame 4 Entry	Frame 5 Entry
Frame No	0	1	2	3	4	5
ITEM NAME	Valve	Pump	Light 1	Light 2	Light 3	Light 4
Item No.	5	6	7	8	9	10
Set Value	999	0	0	0	0	0
COMMUNICATION INFORMATION						
Board No.	3	3	3	3	3	3
Channel No.	0	1	2	3	4	5
Driver VI	Driver 0	Driver 0	Driver 0	Driver 0	Driver 0	Driver 0
EMERGENCY RESPONSE INFO						
Emerg. Sequence Position	1	2	0	0	0	0
Emerg. Value to be set	1 (Drain)	0	0	0	0	0
SYSTEM SHUTOFF INFO						
Shutoff Sequence Position	1	2	3	4	5	6
Value to be set at Shutoff	0	0	0	0	0	0
IDENTIFYING NAME OF THESE SETTINGS	Prob 4 of Chapter 9					
CONSTRAINTS (If none, empty array)						
FRAME NO = 0						
Logical Condition (No name on box)	Empty	Only With	Empty	Empty	Empty	Empty
Item on frame number	Empty	5	Empty	Empty	Empty	Empty

TABLE 9.12

List of Drivers and Directory Location(s)

INSTRUMENT	DRIVER NAME	DIRECTORY LOCATION
Balance 1 Balance 3	Driver RS232 Mettler PJ15	La Cie 200-Q:NIPER Lab WARDEN: Readout & Control Facility: Instrument Drivers:
Balance 2	Driver RS232 Mettler PJ6	ditto
Press. Transd.	Driver RS232 Paroscientific	ditto
Valve, Pump, Light 1, Light 2, Light 3, Light 4	Driver Solenoid ON/OFF EXAMPL	ditto

Example Problem 5—Operation of an Ice Cream Manufacturing Plant

Problem 5 is an example of the versatility of NIPER Lab WARDEN and the user is asked to configure the simplified ice cream plant shown in Fig. 9.21. The plant is a three component plant having a dispensing section, blending/mixing section, and a filling operation. During the filling operation, containers ride a conveyor belt under a dispensing head where the cartons are filled with ice cream. The plant is operated on a batch basis before the operation is switched to a different flavor. For simplicity, each of the flavors are premixed liquids in separate flavor tanks that sit on an electronic balance that has the same electronics as a Mettler PJ15. The pumps have manual speed control that has been set for the specific recipe as well as on/off computer controlled switch. Both the balances and the pumps have VI drivers that have been used in previous problems. The blending tank is refrigerated and has two thermocouples to indicate the temperature at the bottom and to caution when the tank is overfilled. The viscosity of the blend is monitored by a control ammeter that is connected to the motor on the blades of paddle in the blending tank. The ammeter registers 0 to 20 amps. The motor is a 3-phase 480 VAC that at full load pulls 12 KVA. Past experience with this equipment indicates that the mixing is complete when the viscosity of the ice cream causes the ammeter to exceed 9 amps. Only the bottom blade is attached to the shaft, the others are free-rotating. Therefore, even as the tank is depleted the amp load is not much reduced. The ammeter reading should be monitored by the computer and when it exceeds the preset limit of 9 amps will activate a solenoid valve and start pump 5.

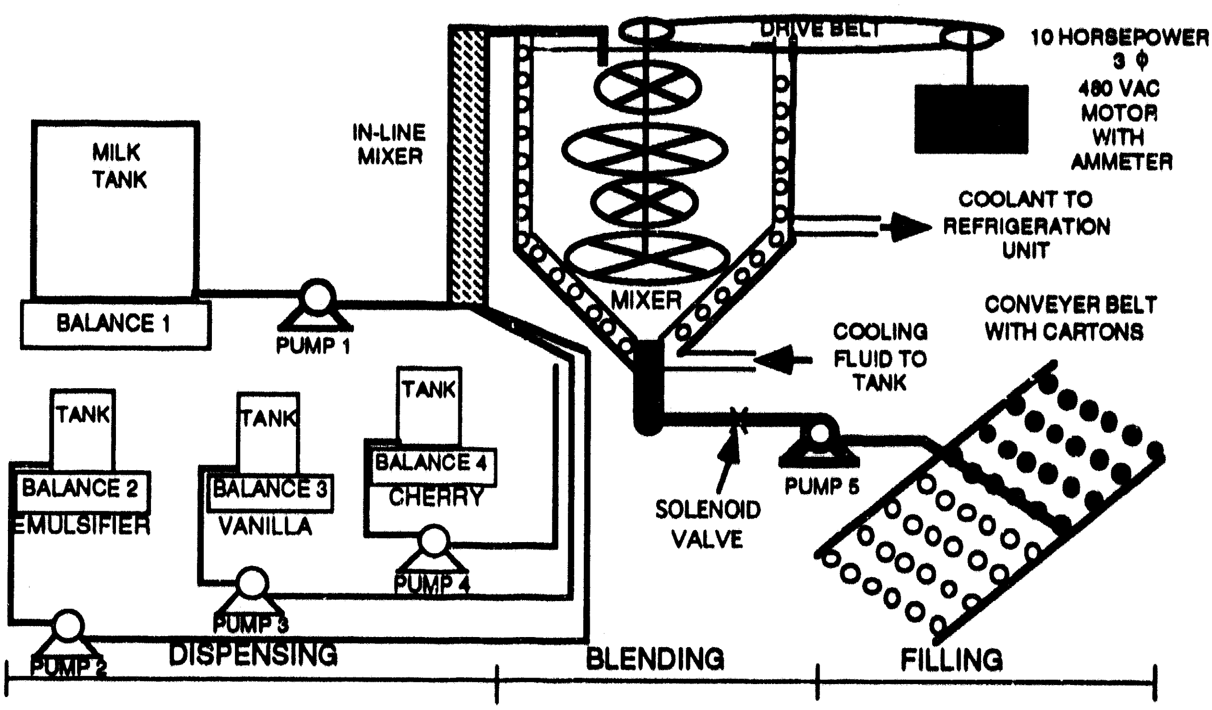


FIGURE 9.21 - Major units in ice cream plant.

The operation is designed to fill six cartons by activating pumps for 10 seconds, then stopped and advance the conveyor belt. This filling cycle continues until all of the ice cream in the mixer tank has been dispensed. The flavor chosen for this batch is cherry-vanilla and with the appropriate manually set pump speeds. The recipe calls for pump 1 to operate 15 minutes, pump 2 for 3 minutes, pump 3 for 5 minutes and pump 4 for 1 minute.

The user is to configure the plant.

Sample Solution for Example Problem 5

It is recommended that problems 1 through 4 be completed before starting this problem. Follow the step-by-step procedure described in Chapter 4, ***To Configure NIPER MAIN FACILITY for Specific Automation Setups.*** The resulting tables from steps 1 through 4 are Tables 9.13 through 9.16, respectively.

In summary, there are four balances, five timers (computer clock), an ammeter, and two thermocouples that function as indicator instruments (i.e. send signals to the computer), and eight control instruments (the valve, motor, conveyer, and pumps 1 through 5) that receive commands from the computer and change their status. The problem requires, at startup, that any filling operation (Fig. 9.21) in progress be stopped (pump 5, valve, and conveyor be turned OFF), and the dispensing (pumps 1 through 4) and blending (motor) operation be started automatically.

As the dispensing is completed, the pumps 1 through 4 are turned off after 15, 3, 5, and 1 minutes, respectively. These timing correspond to each ingredients quota in the ice cream recipe. The timers 1 through 4 (for pump 1 through 4) perform the function of turning them ON/OFF. As each timer exceeds its warning limit, the corresponding pump is turned OFF as a consequence. The four timers use the same driver because they exactly have the same functionality. (An alternate solution for this task may be to manually set the rate of each pump in the right proportion and use only one timer to shut all four pumps simultaneously. The more involved route was chosen for illustrative purposes).

TABLE 9.13

Problem Analysis Work Sheet

STEP 1: Enlist All Automation Instruments In Three Categories (See ***To Classify Instruments On The Basis Of Their Functionality*** for more details):

Indicator instruments:	Balance 1, Balance 2, Balance 3, Balance 4, Ampmeter, Timer 1 through 5, Thermocouple 1 and 2
Control instruments:	Valve, Motor, Conveyer, Pump 1 through 5
Dual-Function instruments:	None

STEP 2: List all automation instruments to be reset at start-up (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump 5 (OFF), Valve (OFF), Conveyer (OFF), Pumps 1 through 4 (ON), Motor (ON)

STEP 3: List all automation instruments to be reset in case of emergency (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump 5 (OFF), Valve (OFF), Conveyer (OFF), Pumps 1 through 4 (OFF), Motor (OFF)

STEP 4: List all automation instruments to be reset at run shut-off (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Same as in STEP 3 above

STEP 5: List all automation instruments to be reset in case of warning (These instruments have to be Control or Dual-Function Instruments and must be included in the listing in STEP 1 above): Pump 1 through 4 (OFF), Pump 5 (ON or OFF), Valve (ON or OFF), Conveyer (ON or OFF)

WARNING CONDITIONS	**CONTROL INSTRUMENTS**	**RESPONSE**
Timer 1 ≥ 15 min	Pump 1	Turn OFF (Reset to 0)
Timer 2 ≥ 3 min	Pump 2	Turn OFF (Reset to 0)
Timer 3 ≥ 5 min	Pump 3	Turn OFF (Reset to 0)
Timer 4 ≥ 1 min	Pump 4	Turn OFF (Reset to 0)
Ampmeter > 9 amp	Valve, Pump 5, Conveyer	Turn ON (Reset to 999)
Timer 5 ≥ 20 min	Pump 5	Turn OFF (Reset to 0)
Wt. Balance 1 ≤ 1000 kgms	None	Audio/Video Warning
Wt. Balance 2 ≤ 100 kgms	None	Audio/Video Warning
Wt. Balance 3 ≤ 100 kgms	None	Audio/Video Warning
Wt. Balance 4 ≤ 100 kgms	None	Audio/Video Warning
Thermocouple 1 ≤ 35^{o} F	Pump 1 through 4	Turn OFF (Reset to 0)

STEP 6: List all automation instruments whose operation is subjected to meeting constraints with object instrument(s) (i.e., the control instruments which can be operated only when the status of target instrument(s) meets the constraint conditions):

CONSTRAINED INSTRUMENT	**OBJECT INSTRUMENT**	**LOGICAL CONSTRAINT**
Pump 5	Valve	Pump Can turn ON "ONLY WITH" Valve ON
	Conveyer	Pump Can turn ON "ONLY WITH" Conveyer ON

TABLE 9.14
Settings for Indicator Instruments

Box Name	Entry For Set #0	Entry For Set #1	Entry For Set #2	Entry For Set #3	Entry For Set #4	Entry For Set #5	Entry For Set #6	Entry For Set #7	Entry For Set #8	Entry For Set #9	Entry For Set #10	Entry For Set#11
Frame No	0	1	2	3	4	5	6	7	8	9	10	11
ITEM NAME	Balance 1 (PJ15)	Balance 2 (PJ6)	Balance 3 (PJ6)	Balance 4 (PJ6)	AmpMeter	Timer 1	Timer 2	Timer 3	Timer 4	Timer 5	Thermocouple 1	Thermocouple 2
Item No.	0	1	2	3	4	5	6	7	8	9	10	11
COMMUNICATION INFORMATION												
Board No.	0	2	2	2	4	N/A (Select Any)	N/A (Select Any)	N/A (Select Any)	N/A (Select Any)	N/A (Select Any)	4	4
Channel No.	0	0	1	2	0	N/A (Select Any)	N/A (Select Any)	N/A (Select Any)	N/A (Select Any)	N/A (Select Any)	1	2
Driver VI	Driver 0	Driver 1	- Driver 1	Driver 1	Driver 2	Driver 3	Driver 3	Driver 3	Driver 3	Driver 3	Driver 2	Driver 2
EMERGENCY & WARNING LIMITS												
Cutoff Higher Limit	0	0	0	0	0	0	0	120	0	0	0	32
Cutoff Lower Limit	500	50	50	50	0	0	0	0	0	0	0	0
Cutoff Accel. (% per min)	0	0	0	0	0	0	0	0	0	0	0	0
Cutoff Decel. (% per min)	0	0	0	0	0	0	0	0	0	0	0	0
Warning Higher Limit	0	0	0	0	9	15	3	5	1	20	0	0
Warning Lower Limit	1000	100	100	100	0	0	0	0	0	0	35	0
Warning Accel. (% per min)	0	0	0	0	0	0	0	0	0	0	0	0
Warning Decel. (% per min)	0	0	0	0	0	0	0	0	0	0	0	0

IDENTIFYING NAME OF THESE SETTINGS	Prob 5 of Chapter 9
COMMON CONFIGURATION (ALL ITEMS)	
Interval (Seconds)	60
Test Status	New:Zero Time
Multiplex Boards	1

ACTIONS DESIRED TO CORRECT WARN. COND.	Set #0	Set #1	Set #2	Set #3	Set #4	Set #5	Set #6	Set #7	Set #8	Set #9	Set #10	Set #11
FRAME NO = 0												
Responding Instrument	Empty	Empty	Empty	Empty	1 (Valve)	3 (Pump 1)	4 (Pump 2)	5 (Pump 3)	6 (Pump 4)	0 (Pump 5)	3 (Pump 1)	Empty
Lower or Higher Limit Status	Empty	Empty	Empty	Empty	Higher Limit	Higher Limit	Highor Limit	Higher Limit	Higher Limit	Higher Limit	Lower Limit	Empty
Reset at % of Value	Empty	Empty	Empty	Empty	999 (ON)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	Empty
FRAME NO = 1												
Responding Instrument	Empty	Empty	Empty	Empty	0 (Pump 5)	Empty	Empty	Empty	Empty	Empty	4 (Pump 2)	Empty
Lower or Higher Limit Status	Empty	Empty	Empty	Empty	Higher Limit	Empty	Empty	Empty	Empty	Empty	Lower Limit	Empty
Reset at % of Value	Empty	Empty	Empty	Empty	999 (ON)	Empty	Empty	Empty	Empty	Empty	0 (OFF)	Empty
FRAM NO = 2												
Responding Instrument	Empty	Empty	Empty	Empty	2 (Conveyer)	Empty	Empty	Empty	Empty	Empty	5 (Pump 3)	Empty
Lower or Higher Limit Status	Empty	Empty	Empty	Empty	Higher Limit	Empty	Empty	Empty	Empty	Empty	Lower Limit	Empty
Reset at % of Value	Empty	Empty	Empty	Empty	999 (ON)	Empty	Empty	Empty	Empty	Empty	0 (OFF)	Empty
FRAM NO = 3												
Responding Instrument	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	6 (Pump 4)	Empty
Lower or Higher Limit Status	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Lower Limit	Empty
Reset at % of Value	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	Empty	0 (OFF)	Empty

TABLE 9.15
Settings for Control Instruments

Box Name	Frame 0 Entry	Frame 1 Entry	Frame 2 Entry	Frame 3 Entry	Frame 4 Entry	Frame 5 Entry	Frame 6 Entry	Frame 7 Entry
Frame No	0	1	2	3	4	5	6	7
ITEM NAME	Pump 5	Valve	Conveyer	Pump 1	Pump 2	Pump 3	Pump 4	Motor
Item No.	10	11	12	13	14	15	16	17
Set Value	0 (OFF)	0 (OFF)	0 (OFF)	999 (ON)	999 (ON)	999 (ON)	999 (ON)	999 (ON)
COMMUNICATION INFORMATION								
Board No.	3	3	3	3	3	3	3	3
Channel No.	0	1	2	3	4	5	6	7
Driver VI	Driver 0	Driver 1	Driver 1	Driver 1	Driver 1	Driver 1	Driver 1	Driver 1
EMERGENCY RESPONSE INFO								
Emerg. Sequence Position	1	2	3	4	5	6	7	8
Emerg. Value to be set	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)
SYSTEM SHUTOFF INFO								
Shutoff Sequence Position	1	2	3	4	5	6	7	8
Value to be set at Shutoff	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)	0 (OFF)
IDENTIFYING NAME OF THESE SETTINGS	Prob 3 of Chapter 9							
CONSTRAINTS (If none, empty array)								
FRAM NO = 0								
Logical Condition (No name on box)	Only With	Empty	Empty	Empty	Empty	Empty	Empty	Empty
Item on frame number	11 (Valve)	Empty	Empty	Empty	Empty	Empty	Empty	Empty
FRAME NO = 1								
Logical Condition (No name on box)	Only With	Empty	Empty	Empty	Empty	Empty	Empty	Empty
Item on frame number	12 (Conveyer)	Empty	Empty	Empty	Empty	Empty	Empty	Empty

TABLE 9.16

List of Drivers and Directory Location(s)

INSTRUMENT	DRIVER NAME	DIRECTORY LOCATION
Balance 2 Balance 3 Balance 4	Driver RS232 Mettler PJ6	La Cie 200-Q:NIPER Lab WARDEN: Readout & Control Facility: Instrument Drivers:
Balance 1	Driver RS232 Mettler PJ15	ditto
Timer 1 through 5	Driver Timer Example	
Ampmeter	Driver Analog Input Example	ditto
Pump 5	Driver Timed ON/OFF EXAMPLE	ditto
Pump 1 through 4, Valve, Motor, Conveyer	Driver Solenoid ON/OFF EXAMPL	ditto

As the ingredients in the blending are whipped by the mixer and also cool down, the viscosity of the ice cream begins to increase which causes the motor to draw more power. As the ammeter shows more than 9 amps, a warning is issued that ice cream is ready to be dispensed, and as a result, the dispensing unit is activated in the following sequence: the valve is turned ON (open), pump 5 is turned ON, and the conveyer is turned ON.

Pump 5 requires a different driver than the ones for pumps 1 through 4 so that it can dispense in a ON-10-seconds/OFF-2-seconds cycles when it is turned ON. The driver is a simple modification of the driver for other pumps and other ON/OFF instruments. The choice to handle the timed-cycle sequence of pump 5 at the driver level was made because of the small time intervals (10 and 2 seconds). For imprecise operation, it can be handled by the same driver. Nonetheless, timer 5 is used to turn this cyclic operation of pump 5 OFF at appropriate time (see next paragraph).

The conveyer is assumed to have an adjustable feed rate, where it can be manually set to advance one step in 2 seconds and stay there for 10 seconds (like in most stepper-motor operations). It also has a manual synch mechanism, where an operator can synchronize it with pump 5 by pressing the "GO" button exactly when the pump has stopped dispensing.

With previous experience, the operator has determined the most efficient batch time to be 20 minutes, i.e. the time it takes to dispense, blend, and fill the entire ice cream batch. What that means is that at 20 minutes, the timer 5 issues a warning that the batch time has ended, and as a result of this warning, pump 5, valve, and the conveyor are turned OFF sequentially. (Alternately, a warning condition such as ammeter ≤ 5 can be set to turn these three instruments OFF because when the mixer tank is emptied and the load on the motor is drastically reduced. This drastic reduction is due to the fact that most high efficiency mixers have fixed blades only at the tip of the shaft—other blades are freely rotating.

Since the operator has selected (from the driver panel) that all the timers (timer 1 through 5) be reset to 0 after 20 minutes, the new cycle begins. As the warning condition is removed from all the control instruments, they go back to their startup values (the NIPER Lab WARDEN program code automatically does that), i.e., pump 5, valve, and conveyor are turned OFF; and pump 1 through 4 and the motor is turned ON. This batch process automatically continues.

Other warning, emergency, and constraints for the process are as follows. When the weight of Milk Tank falls below 1,000 kilograms or the weight of other tanks falls below 100 kilograms, audio/video warning is issued; if no action is taken by the operator and the weight falls below 500 or 50 kilograms, respectively, the instruments are shut down in this sequence: pump 5, valve, and conveyor, pump 1 through 4, and motor. If the cooling in the blending tank is not complete (thermocouple 2> 32° F), the emergency shut down sequence is initiated. If the tank is overfilled (thermocouple 1 ≤ 35° F, indicating that the level has exceeded the thermocouple level), a warning is issued, and if not overridden by the operator, pumps 1 through 4 are turned OFF.

The operator may shut the system down at any time. The selected sequence for normal shut down is the same as the emergency shut down sequence. The only safety constraint in the process is that the pump can not be turned ON unless both the valve and the conveyer are ON.

Problem 6—Operation of a Paroscientific Quartz Pressure Transducer

Problems 1 through 5 were software exercises of increasing complexity in configuring NIPER Lab WARDEN for specific situations. Problems 6 and 7 relate to hardware configuration and are intended to give the user a feel of how instruments interface with the computer. Problem 6 involves interfacing a Paroscientific® quartz pressure transducer with a Macintosh computer.

First, one has to determine how this pressure transducer communicates. The transducer user's manual (Paroscientific, 1990) describes the RS-232 communication protocol, which involves serial communication (a sequence of bits transferred one by one). The manual also contains a circuit diagram explaining how each wire in a 9-pin standard connector (DB9) is connected and how power can be supplied (a simplified diagram is shown in Fig. 9.22). Many computer stores and mail order outlets custom-make these wiring harnesses given the specific instructions, but quite often standard RS-232 connectors work fine.

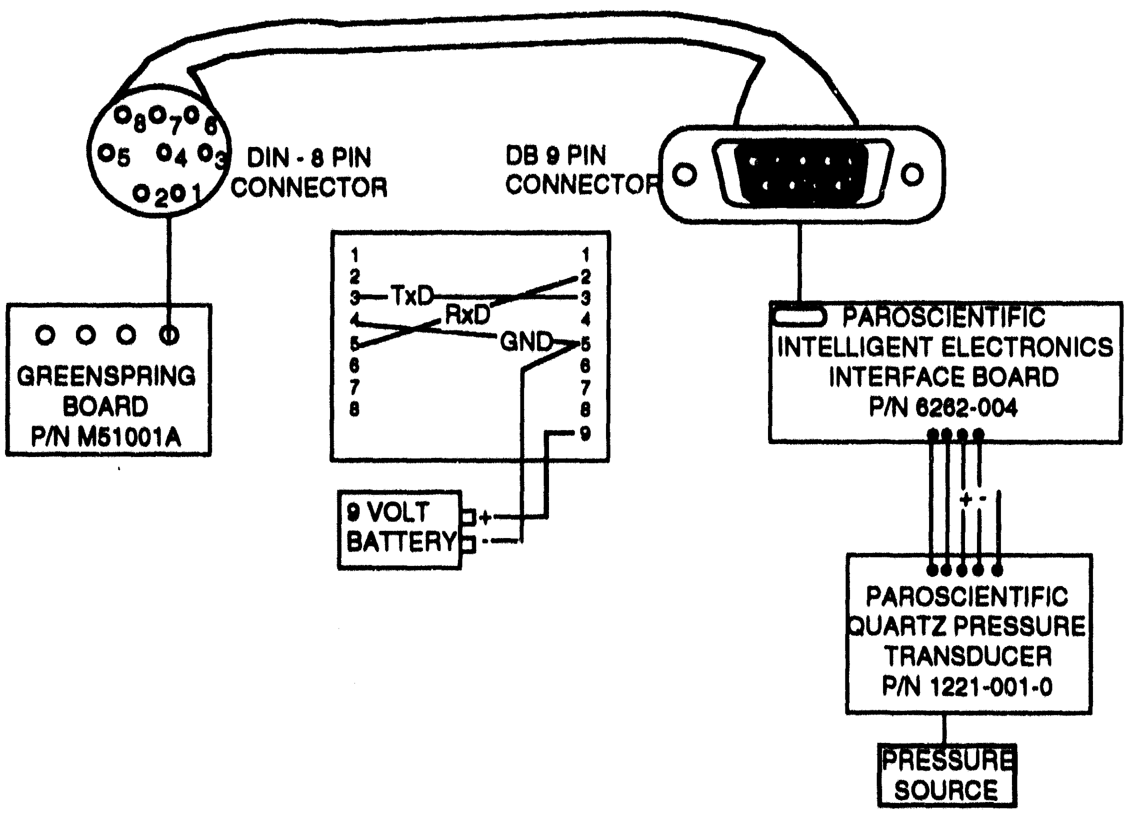


FIGURE 9.22 - Schematic of pressure transducer and computer set-up for problem 6.

The Macintosh comes with a modem port (board number = 0) and a printer mode (board number = 1), both for serial communication. If either port is not being used for interfacing with other instruments or devices, it can be used. If neither is available, then a serial board (e.g., Greenspring® board P/N M51001A) can be installed in one of the extension slots inside the computer. Such boards provide multiple serial outlets similar to the monitor port. Once a suitable port is found and a connector cable is acquired (or made) and connected, the instrument can be physically made to communicate.

After hardware configuration, the next step is to understand how the instrument communicates. Some instruments require that specific strings (commands) be sent to them before they respond (especially multi-function ones). Others continuously send data without receiving any command. The instructions that come with the instrument list the command language (i.e., specific strings for various tasks). The software should be able to send, via the cable, to execute those strings. This task of sending specific commands is handled through a driver program. Similarly, when an instrument sends data back to the computer, it may be in the form of an alphanumeric string that needs to be interpreted according to the instrument instructions in order to get a numeric value. Again, this task is handled by a driver program. NIPER Lab WARDEN relies on these driver programs to read data or write a command to the instrument. NIPER Lab WARDEN's main purposes are to interface with the human operator, manage/supervise (i.e., properly coordinate) all the instruments according to the operator's preset instructions, and present data in a format readily comprehensible to the operator.

Problem 7—Operation of a High Voltage Electrical Heater

This problem, like problem 6, is a hardware configuration problem. The problem is to control an electrical resistance heater. A Eurotherm® temperature controller, a Eurotherm Thyrister, and the resistance heater must be controlled to regulate the temperature using a the computer. One way for the computer to communicate with the temperature controller is through the National Instruments NB-DIO-24 board that is connected to a high voltage relay board (Gordos® P/N PB-24). This relay board in turn controls the temperature controller and a higher voltage relay (Eurotherm Thyrister). A schematic of the major components for this problem is shown in Fig. 9.23.

The hardware wiring shown in Fig. 9.23 requires a 0-3 VDC signal from the National Instrument's board driver relays on the Gordos board in turn during higher and higher voltage relays. An external 5VDC to energize the Gordos rack-mounted relay board, which in turn will produce 110 VAC to operate the Eurotherm controller. The Eurotherm Thyrister requires an external 220 VAC source. A thermocouple is also needed in order to monitor the temperature. The manual supplied with the Eurotherm controller provides information on how to wire the controller to the Thyrister. A driver program needs to be written which can be integrated with NIPER Lab WARDEN (not shown).

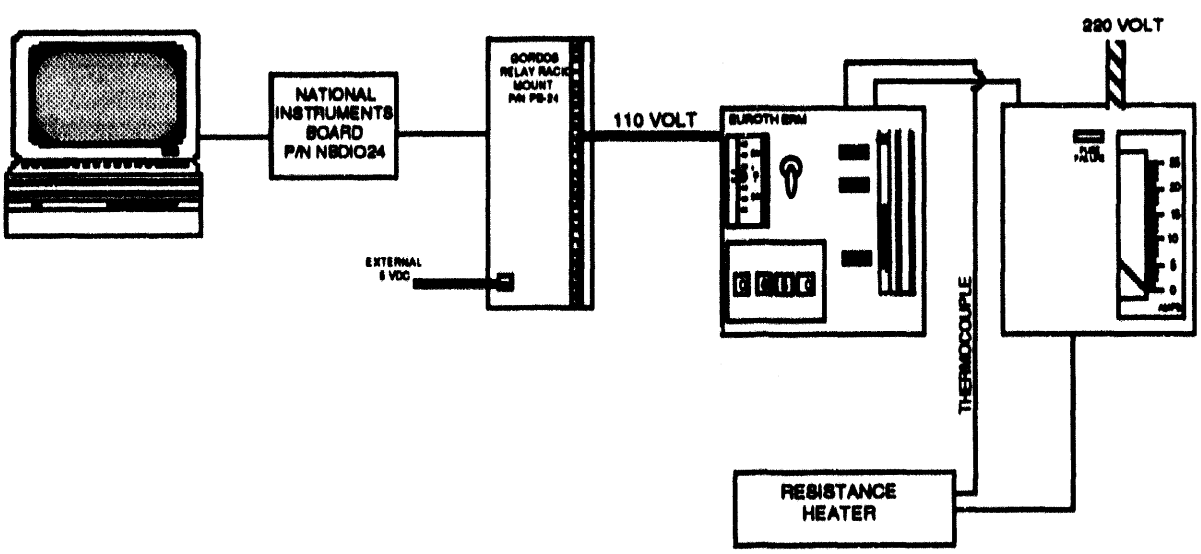


FIGURE 9.23 - Schematic of control of a high voltage electrical heater.

BIBLIOGRAPHY

Circle Seal Controls, 1988. *Seven SV10-40 and SV400* Series, *2- and 3-Way Leakproof Solenoid Valves, Circle Seal Controls*, 1111 N. Brookhurst, Anaheim, CA, 92803, January.

Eurotherm Corp., 1987. *Products Catalog 1027-A*, Eurotherm Corp., 11485 Sunset Hills Road, Reston, VA, 22090, January.

Eurotherm Corp., 1982. *Eurotherm Operation and Maintenance Manual For Digital Temperature Controller Model 918*, Eurotherm Corp., 11485 Sunset Hills Road, Reston, VA, 22090-5286, July.

Gordos, 1993. *Specifications for PB24 Relay Board.* Newark Electronic, Part No. 21F858 Mounting Board PB-24, Newark Electronics, Tulsa, OK, August.

Greensprings Computer Inc., 1990. *Operation Manual for Multiport (4) Serial Board, P/N M51001A*, Greensprings Computer Inc., 1204 O'Brian Drive, Menlo Park, CA, 94025, January.

Mettler, 1990. *Mettler Balance PJ15 User's Manual*, Mettler Instrument Corp., Box 71, Hightstown, NJ, 08520-9944, April

National Instruments Corp., 1991a. LabVIEW 2—*Getting Started Manual*, Part No. 320246-01, Austin, TX, April.

National Instruments Corp., 1991b. *Training In-Depth Course on LabVIEW 2*, Version 1.4, Part No. 776393-01, Austin, TX, April.

National Instruments Corp., 1991c. *LabVIEW 2—User Manual*, Part No. 320244-01, Austin, TX, September.

National Instruments Corp., 1992. NB-DIO-24—*User Manual*, Part No. 320094-01, Austin, TX, October.

Paroscientific, Inc., 1990. *Digiquartz Precision Pressure Instruments, Programming and Operation Manual*, Document No. 8107-001, 4500 148th Avenue N.E., Redmond, WA 98052, August.

☆U.S. GOVERNMENT PRINTING OFFICE: 1994 - 860-030/00011

DATE
FILMED
6/10/94

END