THE COLLEGE OF
ENGINEERING

Labview on a Sun Microsystems Workstation

ME106 Mechatronics
Final Project Report

Prepared For: Dr. Burford Furman
Revised: 05/12/1998
Author:Thomas Stewart
Team Members: Thomas Stewart, Bala Durairaj, Jimmy Lau
Email the professor: *bjfurman@email.sjsu.edu*

Table of Contents

Abstract:

Sun Microsystems purchased more than $10,000 in hardware and software to automate the time and manpower intensive process of collecting thermal data with thermocouples. Sun did not have the internal expertise in the data acquisition software Labview, and therefore the automated thermal station was never completed. The purpose of this project for ME106 at San Jose State University was to learn about the Labview program using tutorials and manuals, solve the problem for Sun, and demonstrate the ability to acquire data directly to the Sun workstation using the available hardware.

Back to table of contents

Acknowledgments:

This project would not have been possible without the support and assistance of several individuals at three companies.

1. **Jennifer Zahm** at Sun Microsystems for providing background material, access to the hardware, and bringing the team up to speed on Sun's problem.
2. **Robyn Thompson** at National Instruments, the company that sells the Labview graphical programming tool, for her extra effort on the phone and over email. Her support enabled us to finish this project by explaining key parts of the process, providing the final graphical instrument drivers, and testing them at her office on a Sun workstation to ensure they would work for us.
3. **Charles Fustos** at IOTech for his assistance and support to help us understand the IOTech Tempscan datalogger and get the labview software talking to the datalogger.

Back to table of contents

Executive Summary

This project had three main hurdles the team had to overcome to solve the problem.

1. Understanding the basics of Labview including wiring diagrams, user interface design, and how to build and use virtual instruments (vi's).
2. Configuring and installing the Sun workstation on the corporate network, using unix to manipulate and install the instrument driver vi's, and overcoming the compatibility issues between Labview versions, IOTech driver versions, and Solaris operating system incompatibilities.
3. Configuration of the hardware and GPIB interface controls to successfully connect the Sun workstation to the datalogger.

The first problem was solved by obtaining tutorials and manuals from the Mechatronics lab and Sun Microsystems. For the first part of the semester, the entire team worked in parallel to learn about Labview, run practice exercises, and understand the fundamentals of labview. Our team was able to work in parallel at Sun, by obtaining codes to the ME Laboratory from Dr. Furman, and working on PC's at home.

The Sun workstation problems were solved by systematically identifying the problems and continuously moving the project forward during the semester. At the completion of the project, the following configuration was successful:

- Solaris 2.5.1 (SunOS 5.5.1)
- Labview 4.1 upgrade
- Reconfigured IOTech instrument drivers
- Networking software NIS (not NIS+)
- Creating a network node called "labview1" on the Sun network

The hardware problem proved to be the easiest to solve, but it only turned out that way in hindsight. Simply setting the dip switches on the back to address 1 allows the Labview vi to acquire data through gpib1 menu selection. Proper setting of triggers and the use of the trigger command in the vi finally allowed data to be displayed on the Sun workstation.

Back to table of contents

Introduction:

Sun Microsystems had purchased approximately $10,000 in hardware and software to automate the thermal testing at Sun using dataloggers and thermocouples. Sun did not have the internal expertise in Labview, and had a contractor bid on this project. The bid was $10,000, and Sun was hesitant to spend another $10,000 on this project without understanding the return on the investment (ROI). Therefore when this project was presented to Sun by SJSU students, it was accepted since spending $0.00 was well within budget. Sun had only to provide access to the laboratory and the hardware involved in the project.

The **functional specifications** for this project were as follows:

1. Sun Microsystems Workstation
2. National Instruments GPIB s-bus adapter card
3. IOTech Tempscan 1100 datalogger
4. J-type thermocouples
5. GPIB cable
6. Labview 4.1
7. Data to be transferred from the datalogger to the Sun, presented in a graph of Temperature vs. Time, respond correctly to increases or decreases in the temperature, and be accurate to within 10% FSO.

Labview is a computer program development application similar to programming languages such as C and BASIC. Instead of using text-based languages to create lines of code, Labview uses graphical symbols (icons) to describe programming actions. Labview uses symbols, terminology, and ideas that are familiar to scientists and engineers. Labivew also contains features to perform data acquisition, instrument control, data analysis, data presentation, and data storage. The first two aspects of Labview, data acquisition and instrument control, are the focus of this project. The demonstration to the class will detail how to interface the VI with a spreadsheet for data storage, and show samples of data analysis and presentation by Bala and Jimmy.

Labview is programmed in a language called "G", and the programs produced are called virtual instruments (VI's). They are called virtual instruments, because the buttons, controls, knobs, and features can be created to resemble an actual piece of test equipment. The operation of the VI, or flow control, is directed by the block diagram. This block diagram is a wiring diagram that provides the structure like the C-programming features of "while", "for", and "if", but in a graphical representation. VI's use a hierarchical structure such that one VI can be made up of several smaller VI's, called sub-VI's. Each sub-VI can be designed to perform a specific function, which simplifies debugging and allows for the reuse of VI's for other instruments.

The use of Labview is prevalent in the high tech industry. Allowing an easy to use front-end, the capability to save data into spreadsheets, preventing transcription errors, and saving hours of manpower per test is why Labview is used in industry.

Back to table of contents

The Solution:

At the beginning of the project, we began using the tutorials and reference manuals that came with Labview. We quickly realized that there were discrepancies between the manuals and the program, and found that Sun had upgraded Labivew to version 4.1. The manuals were version 4.0, so we then obtained the newer manuals for 4.1

We continued learning about Labview basics individually, and in parallel each team member took responsibility for a major portion of the project as described below.

Team Member	Primary Responsibility	Secondary Responsibility
Bala	creating a functional VI	documenting the flow of the VI's
Jimmy	designing the possible Labview Lab for ME106	learning how to use Labview and build VI's
Thomas	build & configure the Sun system, work with National Instruments & IOTech to obtain the device and instrument drivers, and make the data acquisition system work with a Labview VI	learning how to use Labview and build VI's

Once the team became familiar with Labview and the basic function of the language, we realized that the instrument drivers for the IOTech tempscan 1100 datalogger were incompatible with the upgrade version 4.1 of Labview. The original Tempscan drivers were based upon 16 bit operation, and Labview 4.1 was a 32 bit program. We obtained new instrument drivers fro IOTech, but these were windows based ZIP files that were unusable for the Sun workstation. We then worked with National Instruments to determine our options and how we could complete the project. Although they offered VI converter programs for about $300, we were fortunate to obtain help from Robyn Thompson in technical support for National. Robyn took the 32 bit files from IOTech I forwarded to her via email, and took out the labview LLB files and made them function on a Sun workstation

at her facility in Texas. She then packaged the utilities in a ZIP file and emailed them to Thomas for installation. After several difficult days of installation problems, we managed to get the drivers installed and working with the VI. The final problem was figuring out how to use the supplied VI's that did not come with documentation. Once this was completed, were were able to obtain data from the datalogger and display it on the screen in the Labview VI.

In addtion to getting the Sun project functional, Bala created several VI's and documented these results with descriptions and flow charts shown in section one of the hardcopy attachments. See Dr. Furman for these hardcopy files and copies of the electronic versions of the VI's created.

The final part of this project was completed by Jimmy. Jimmy outlined a possible Laboratory 6 for the ME 106 class to teach future students the fundamentals of Labview we learned this semester. In the current laboratory manual, there is no laboratory 6, so the lab procedure shown in section 2 of the attachment hardcopy could be used without changing the numbering of the current lab manual.

Back to table of contents

Results:

The following list details the accomplishments and results of this project:

1. The team learned the fundamentals of labview and became familiar with this programming tool.
2. The Sun workstation and data logger setup is now functional for Sun Microsystems to use to capture thermal data.
3. Four virtual instruments were created and documented for use by Sun and/or San Jose State in future laboratory assignments.
4. A laboratory procedure is completed that could be the starting point for students and fill in laboratory number 6 for the ME106 course.
5. This report describing the basics of Labview was written in html to be made available on the ME106 home page for use and reference by future students.

Back to table of contents

Conclusions:

The following conclusions were made by the team regarding labview and over the course of this project:

- Upgrades to key programming tools like Labview should be considered careful with respect to the compatability of dependent software, like the IOTech instrument drivers.
- Labview is a powerful and easy to use tool once the initial steep learning curve is overcome. Labview takes a bit of time to become familiar with, and continued use to become proficient in programming and automating test equipment.
- It is more difficult to integrate software programs on a Sun Workstation than on a Windows-based PC because vendors are more familiar with PC's due to higher sales volume and customer requests.
- An introduction to Labview could be an important addition to the ME106 laboratory section of the course.
- Project reports in engineering should all by written in html so that students can leverage previous results, learn from each other, and expand on the work done by students in previous semesters. This project could be expanded and extended if taken from this point forward in a future semester.

Back to table of contents

Recommendations:

We recommend that Sun take what we have completed in this project and automate the datalogger and other equipment in their environmental laboratory. The next step would be to take the wiring diagram from the LLB from IOTech and National, and combine those features with the VI's created by this team to make the program easier to use, and more compatable with Sun's process.

Back to table of contents

Proposed Lab Experiment:

Find attached in hardcopy format the proposal for a laboratory experiment for ME106 to introduce labview to students. See section 2 of the hardcopy attachment. Back to table of contents

---