Lab 04: Introduction to LabView

EG1003 Section G1

Alexander Lopez-Matos

Lab Partner: Kevin

Date of Experiment: September 29, 2014

Date due: October 5, 2014

**Abstract**

The objective of this lab Introduction to LabView is to create three different systems using the computer application LabView. The first application is a digital calculator, the second is a building lighting system, and the third is a thermal control system that dynamically adjusts itself based on the outside temperature.

LabView allows us to simulate instruments that could be used in a laboratory setting and allows us to collect data from the real world and bring it into a computer. LabView also introduces a “graphical programming language” which differs from traditional text based programming languages in that all the logic is visual.

**Introduction**

LabView allows for the creation of graphical programs named “virtual instruments” or VI for short. These Virtual Instruments emulate real laboratory devices. To create a Virtual Instrument in LabView, a “front panel” and a “back panel” are required. The front panel of the graphical application is where all input and output is taken care of. For example, on the front panel of a program, you could have a text box that takes in a number of a string of characters. The back panel is where the logic of the application is located. On the back panel, you could acquire a number and then perform a mathematical operation on it such as square root, and then either use that square root somewhere else in the logic or display it right back to the user.

Front panels are constructed by using “controls” and “indicators”. Controls on the front panel are dedicated strictly for input. In other words, controls are things such as a thermometer or even just a text box to retrieve data from the user. Indicators are for outputting information to the user. When the back panel processes the information given to it from the controls, it displays information back to the users by using indicators such as a text or number view or things like radio buttons.

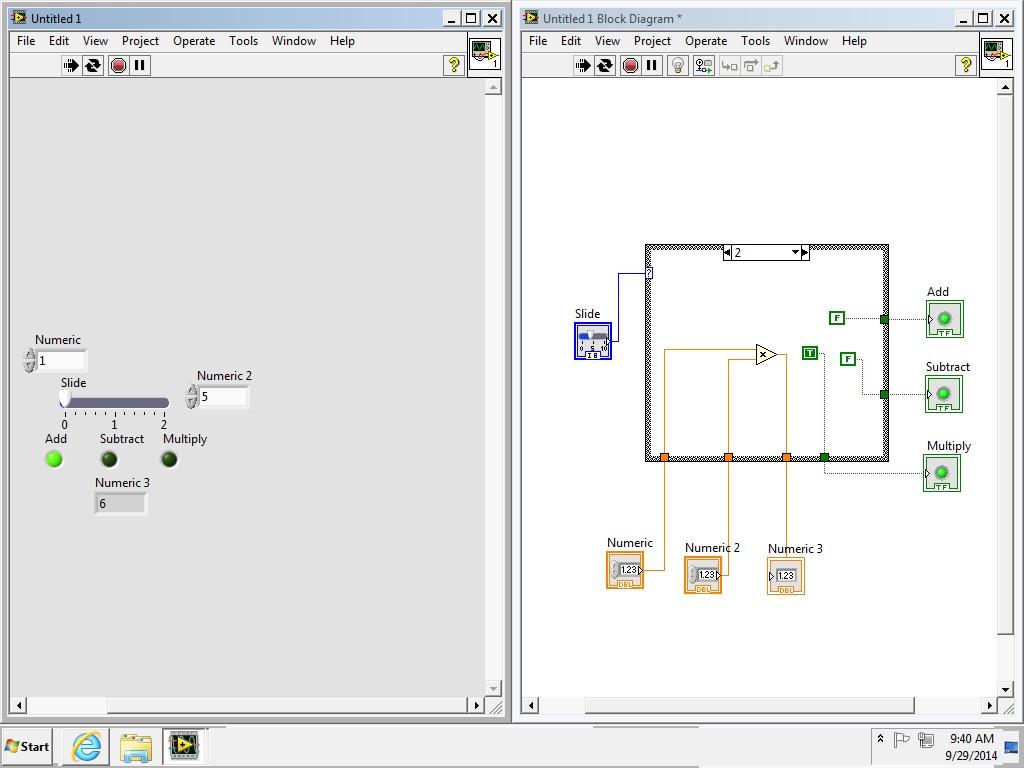
The back panel of a LabView program is also called a “block diagram”. The block diagram contains the “source code”, or the logic, of what the program is meant to do. Within this logical flow structure, there are components called “structures” and “functions”. Structures interact with functions to process data and then display it back to the user by using an indicator on the front panel.

**Procedure**

The procedure for creating the Heating and Cooling Virtual Instrument is as follows. Create a heating and cooling virtual instrument that reads the temperature and turns on the aur conditioning if the temperature reading is above 80 degrees Fahrenheit and enables the heater if the temperature is less than 60 degrees Fahrenheit. Add three indicators on the front panel to represent the AC, the heater, and the manual option. Create three switches to be able to control the AC and heat manually. Set the thermometer to 70 degrees Farenheit. Construct the back panel with boolean statements in order to control the heat and ac accordingly. Include the pre-written “sub” VI’s to interact with the NI ELVIS board. Connect the output tunnels with the heat and cooling. Ask the TA’s to review the modifications and then test the program with the NELVIS board.

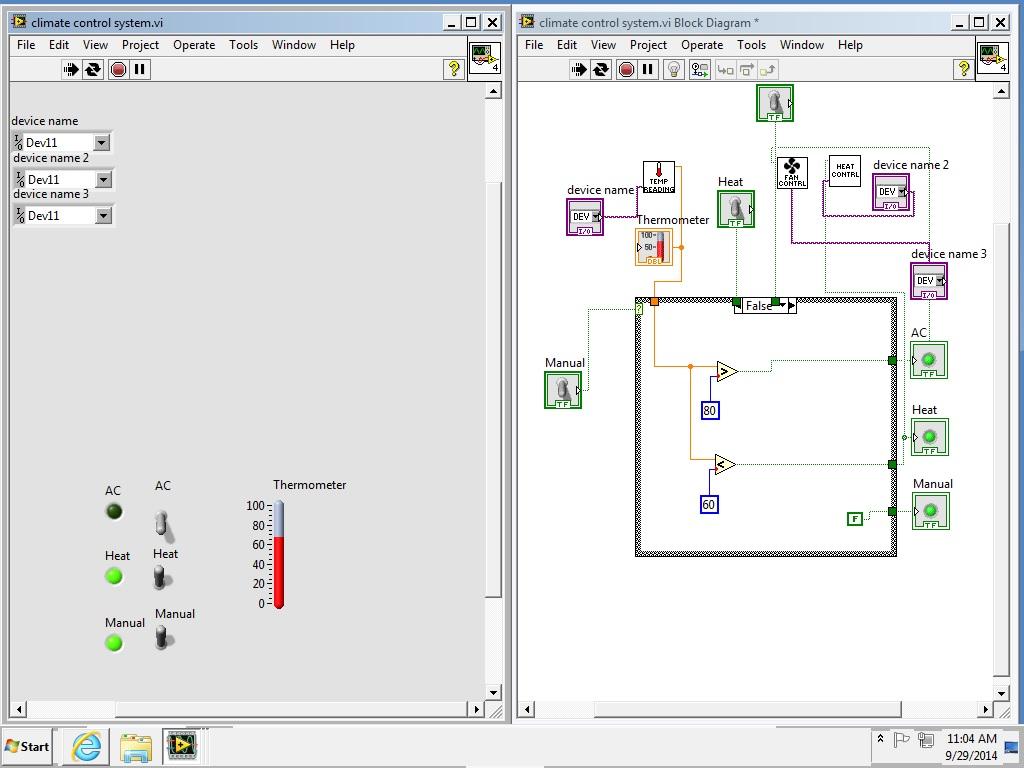
**Data/Observations**

The first design was a simple calculator. The calculator, as shown in figure 1, takes in two numeric inputs and then performs either addition, subtraction, or multiplication on the numbers, and then displays the answer in the third text box. This simple program uses a case structure to determine which operation needs to be performed when the slider is moved either left or right. When the slider is on 0, addition is performed on the two inputs. When the slider is on 1, subtraction is performed, and when it is on 2, multiplication is done. Although this program was the easiest out of the three designs to create, it taught a great deal about the inner workings of a LabView virtual instrument.

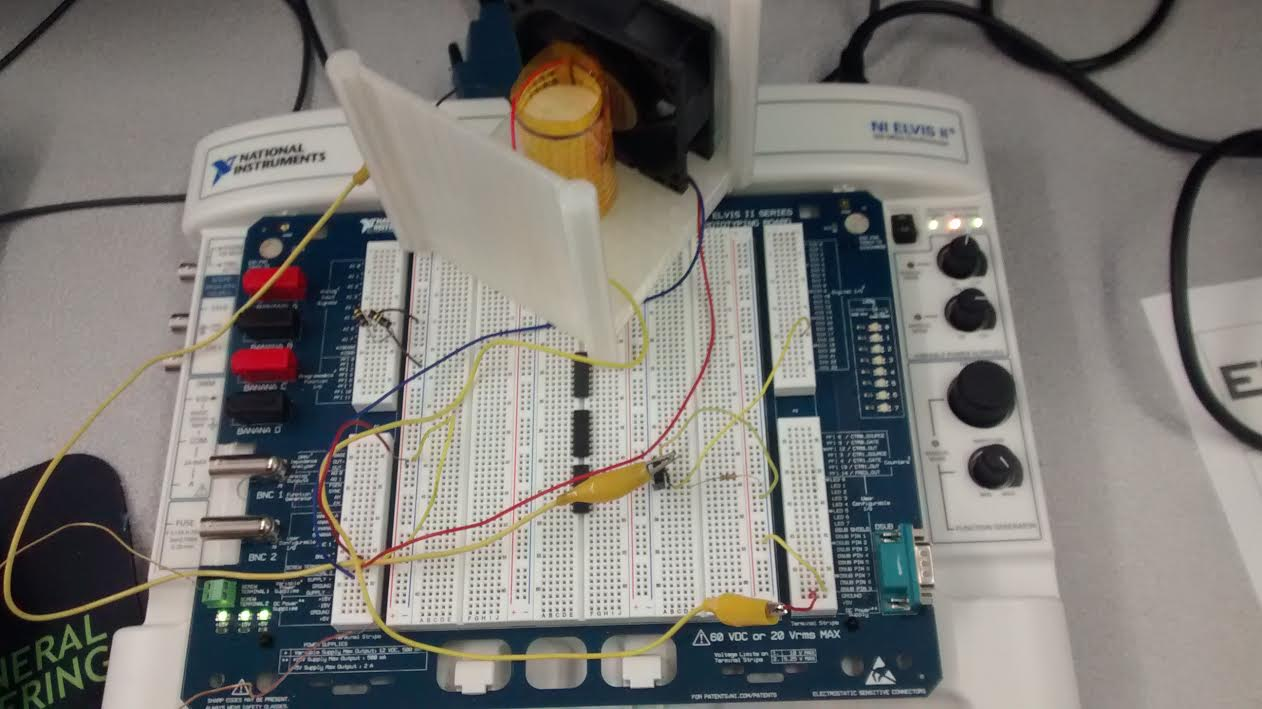
**Figure 1 -** Calculator

The second design, as shown in Figure 2, was the cooling and heating system. The heating and cooling system has two modes, automatic and manual. While in manual, air condition or heat can be turned on regardless of the current temperature. When in automatic mode, if the temperature reaches 80 degrees Fahrenheit or higher, the air condition is turn on, and if the temperature drops below 60 degrees, the heater is turned on. Once the initial program was created, the program was modified to then work with the NEVLIS board, shown in figure 3.

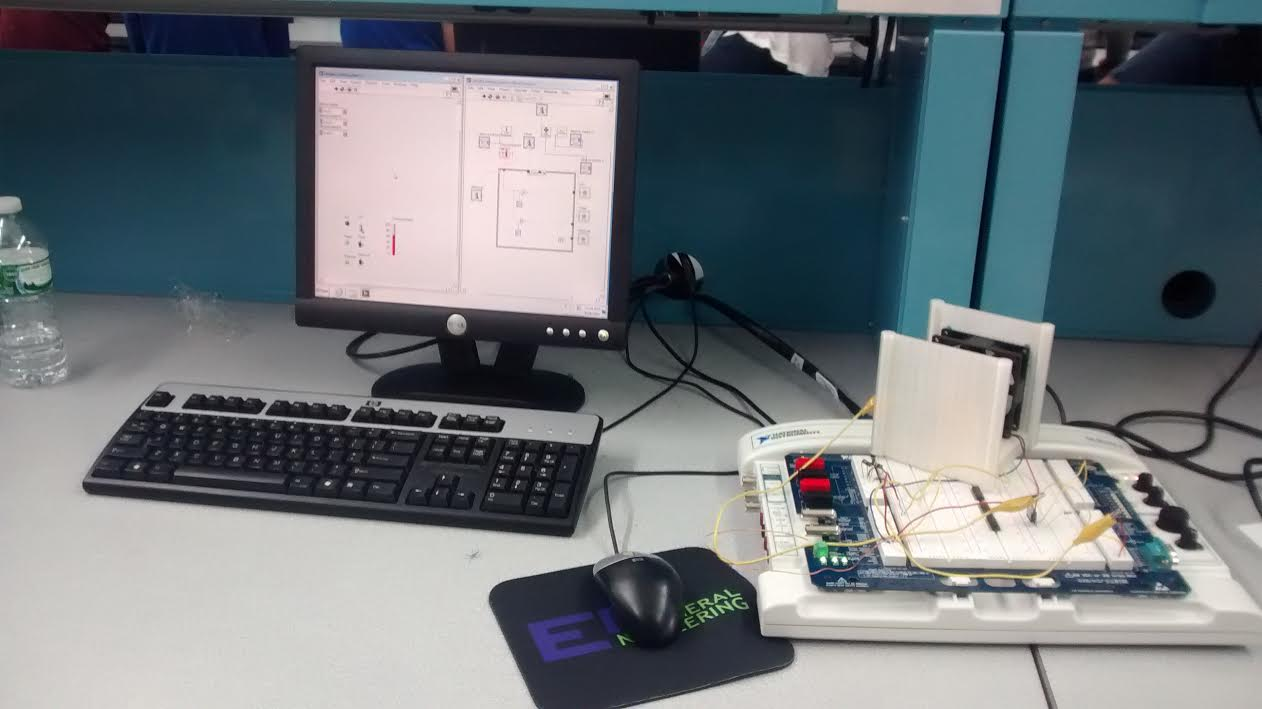
F**igure 2 -** Improved design



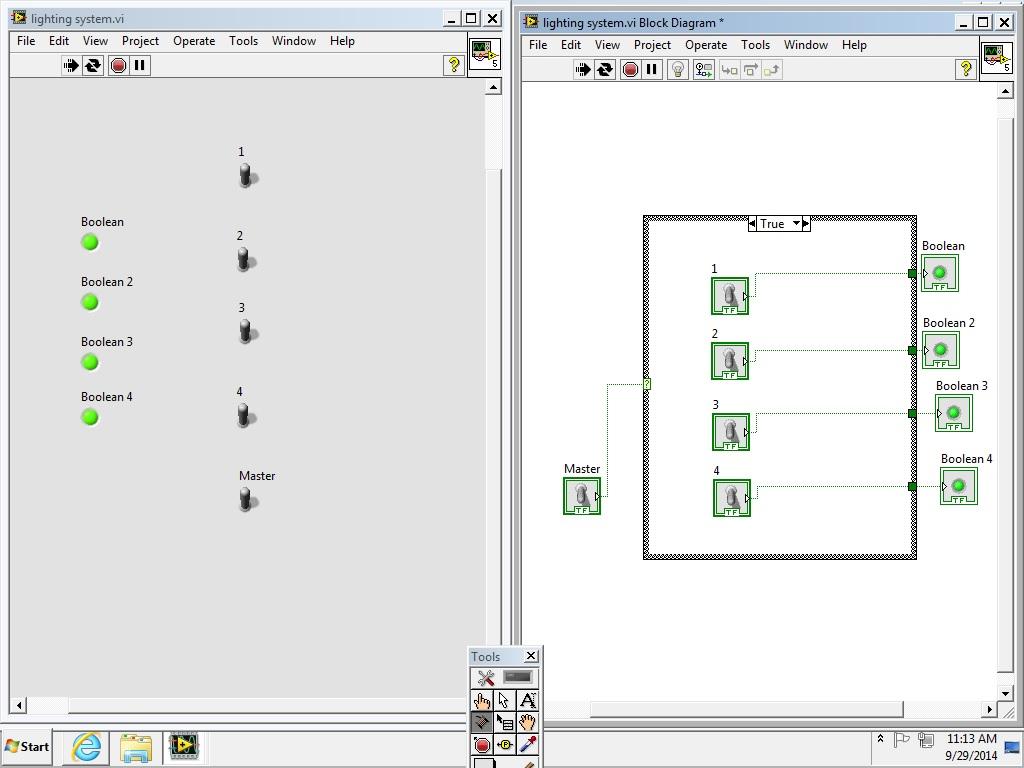
F**igure 3 -** N ELVIS board



The NELVIS Board was then connected to the program and activated the heater and cooler attached to it according to the temperature reading. This design was significantly harder than the calculator, but it allowed our team to grasp the power of LabView or similar development environments in the read world where data needs to be collected from the physical and converted into the mechanical and computer logical.

F**igure 3 -** 4 ELVIS board connection to computer and LabView

The third design was a building lighting system, shown in figure 5. The virtual instrument consisted of four lights and with four corresponding switches and one master switch. When the switches were changed, the light that connected to the switch was turned on or off.

F**igure 5 -** Lighting system

**Discussion/Conclusions**

In this lab, our team learned that LabView is a powerful engineering development tool that has many different applications in a laboratory setting. Through the power of LabView and the NELVIS board, creating programs that take effect in the real world is possible. The Cooling and Heating system in particular clearly demonstrates the power of the tools used in this lab because it dynamically adjusts to real world environmental variables. Creating something similar in a traditional programming language or development environment would be incredibly difficult, as communicating between physical devices and computer language is not an easy task. LabView eases that communication and allows engineers to focus primarily on the logic of the Virtual Instrument they are trying to create, rather than focusing on attempting to interface a real world device.

Work Cited

"EG Manual" *EG1003*. N.p., n.d. Web. 28 Sept. 2014. (manual.eg.poly.edu/index.php/Main\_Page)

