**Department of Electrical Engineering and   
Computer Science**

**Faculty Member:** Dr. Shahzad Younis  **Dated:** 14/09/2022

**Semester:** 5th **Section:** BEE 12C

**EE-383:** **Instrumentation and Measurements**

Lab 1: Introduction to LabVIEW

Lab Instructor: Mr. Ali

Group Members

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Reg. No** | **Conduct of**  **Experiment** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics**  **and**  **Safety** | **Individual and Teamwork** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
| Danial Ahmad | 331388 |  |  |  |  |  |
| Muhammad Umer | 345834 |  |  |  |  |  |
| Tariq Umar | 334943 |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Table of Contents**

[2 Table of Figures 3](#_Toc114349493)

[3 Introduction to LabVIEW 4](#_Toc114349494)

[3.1 Objectives 4](#_Toc114349495)

[3.2 Equipment 4](#_Toc114349496)

[3.3 Introduction to LabVIEW 4](#_Toc114349497)

[3.4 Lab Instructions 4](#_Toc114349498)

[4 Lab Tasks 5](#_Toc114349499)

[4.1 Adding two numbers 5](#_Toc114349500)

[4.2 Adding two randomly generated numbers 5](#_Toc114349501)

[4.3 Continuously running a program 5](#_Toc114349502)

[4.4 Usage of WHILE loop 6](#_Toc114349503)

[4.5 Conversion from WHILE to FOR loop 6](#_Toc114349504)

[4.6 Using FOR loop as a conditional 7](#_Toc114349505)

[4.7 Displaying a string as per the condition 7](#_Toc114349506)

[4.8 Using CASE STRUCTURE to turn LED on as per the condition 8](#_Toc114349507)

[5 Conclusion: 8](#_Toc114349508)

# Table of Figures

[Figure 4.1‑a: Block Diagram 4](#_Toc114676108)

[Figure 4.1‑b: Front Panel 4](#_Toc114676109)

[Figure 4.2‑a: Block Diagram 5](#_Toc114676110)

[Figure 4.2‑b: Front Panel 5](#_Toc114676111)

[Figure 4.3‑a Controls 5](#_Toc114676112)

[Figure 4.3‑b Simulation 5](#_Toc114676113)

[Figure 4.4‑a: Block Diagram 6](#_Toc114676114)

[Figure 4.4‑b: Simulation 6](#_Toc114676115)

[Figure 4.5‑a: Context Menu 6](#_Toc114676116)

[Figure 4.5‑b: Block Diagram 6](#_Toc114676117)

[Figure 4.6‑a: Context Menu 7](#_Toc114676118)

[Figure 4.6‑b: Conditional Terminal 7](#_Toc114676119)

[Figure 4.7‑a: Block Diagram 7](#_Toc114676120)

[Figure 4.7‑b: Simulation 7](#_Toc114676121)

[Figure 4.8‑a: Block Diagram 8](#_Toc114676122)

[Figure 4.8‑b: Simulation 8](#_Toc114676123)

# Introduction to LabVIEW

## Objectives

* Introduction to LabVIEW
* Designing a simple VI
* Introducing Control and Simulation Module in LabVIEW

## Equipment

Software

* *LabVIEW*



## Introduction to LabVIEW

Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) is a platform and development environment for a visual programming language from National Instruments. It is a very popular software tool among electrical engineers. LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications.

There is also another important difference between text-based programming languages and LabVIEW. In most text-based programming languages instructions determine program execution. LabVIEW uses dataflow programming, where the flow of data determines execution. This point will become clearer as you will work in LabVIEW.

## Lab Instructions

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

* Lab objectives
* Results (Graphs/Tables/Pictures) duly commented and discussed
* Conclusion

# Lab Tasks

## Adding two numbers

|  |  |
| --- | --- |
|  |  |
| Figure ‑: Block Diagram | Figure ‑: Front Panel |

**Comments:** In this task we used two numeric numbers and one indicator. The two number were added by the addition block and the result is shown on the display.

## Adding two randomly generated numbers

|  |  |
| --- | --- |
|  |  |
| Figure ‑: Block Diagram | Figure ‑: Front Panel |

**Comments:** In this task we used two random number generators and one indicator. The two generated numbers are added by the addition block and the result is shown on the display. This random generated has a range from 0-1.

## Continuously running a program

|  |  |
| --- | --- |
|  |  |
| Figure ‑ Controls | Figure ‑ Simulation |

**Comments:** In this task we used the in-built simulation bottom to run a specific piece of block continuously. This was achieved using the continuous run  symbol.

## Usage of WHILE loop

|  |  |
| --- | --- |
|  |  |
| Figure ‑: Block Diagram | Figure ‑: Simulation |

**Comments:** In this task we make use of loops to run a task continuously. A WHILE loop or a FOR loop can be used for this purpose. For the present case, we are using a WHILE loop. In a while loop we give the stop button any condition, when the condition is true, it stops the loop.

We also inserted the delay block , which adds a delay *(in ms)* to an iteration based on the constant fed to it. Here it is causing a delay of 500ms or 0.5s.

## Conversion from WHILE to FOR loop

|  |  |
| --- | --- |
|  |  |
| Figure ‑: Context Menu | Figure ‑: Block Diagram |

**Comments:** In this task, we are making use of a FOR loop. A FOR loop can be used instead of a while loop to perform a task for as many number of times we want by assigning a value to the variable N in the top left corner of the block.

## Using FOR loop as a conditional

|  |  |
| --- | --- |
|  |  |
| Figure ‑: Context Menu | Figure ‑: Conditional Terminal |

**Comments:** To add terminating conditions to a FOR loop in addition to the iteration based counter on the top left corner of the block, we right click and select the conditional terminal. In Figure 6-b, we can make use of the stop button to break a FOR loop upon clicking it.

## Displaying a string as per the condition

|  |  |
| --- | --- |
|  |  |
| Figure ‑: Block Diagram | Figure ‑: Simulation |

**Comments:** In this task, we are adding two numbers and comparing it with 1. Based on the result, we are displaying a specific string on the String Indicator. A logical block (Greater Than) and a decision block (Selector) is used for this purpose. If the output of the logical block is 1, then the string on the upper side of the selected and in case of 0, the one on the lower side is selected.

## Using CASE STRUCTURE to turn LED on as per the condition

|  |  |
| --- | --- |
|  |  |
| Figure ‑: Block Diagram | Figure ‑: Simulation |

**Comments:** The decision block can be replaced by a case structure block. In the case structure block there are two areas, one for true and other for false. Based on the condition given to the case block, either the block is true or false are executed and the result can be moved out of the case block for further computation. For the present case, we are making an indicator ON or OFF based on the output.

# Conclusion:

In this lab, we have familiarized ourselves with LabVIEW; performed basic functions like numeric addition and logical decision making. Random numeric values were generated using the random numeric number generator and a numeric display block was used to show the numbers generated. Along with this, the string indicator was used to show different strings based on some conditions. We also explored WHILE loop, FOR loop, and used them perform a task repeatedly. Case structure block was also used to do different tasks based on the condition (Ture of False) given to it.

It was a basic lab which introduced us to the software and some of its basic functions and how to use them in future labs and projects.