***INTERNSHIP REPORT***

***submitted in partial fulfilment of the requirements for the Award of Degree of***

***BACHELOR OF TECHNOLOGY***

***in***

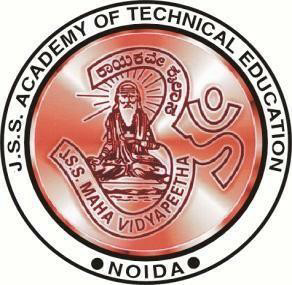
***ELECTRONICS AND COMMUNICAION ENGINEERING***

***in***

## National Instruments

SUBMITTED BY: Aditya prakash Dubey

(ROLL NO:1809131011)



***DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING***

Mrs.Chaya Grover

#### JSS ACADEMY OF TECHNICAL EDUCATION C-20/1 SECTOR-62, NOIDA

2021- 2022

## CERTIFICATE



**TABLE OF CONTENTS**

| **Chapter No.** | **Topic** | **PAGE No.** |
| --- | --- | --- |
|  | INTERNSHIP CERTIFICATE | 2 |
|  | ACKNOWLEDGEMENT | 4 |
|  | ABSTRACT | 5 |
|  | LIST OF FIGURES | 6 |
| CHAPTER 1 | INTRODUCTION | 7 |
| CHAPTER 2 | WHAT I HAVE LEARNED | 17 |
|  | REFERENCES | 29 |

## ACKNOWLEDGEMENT

With immense pleasure I, Aditya prakash Dubey presenting Internship Seminar report as part of the curriculum of B. Tech, Fourth year (seventh semester).

I am extremely grateful to Prof. Gurulingappa M. Patil, Principal JSS Academy of Technical Education, Noida and Dr. Arun Kumar G., Head of Department, Department of Electronics & Communication Engineering, for providing all the required resources for the successful completion of my seminar. My heartfelt gratitude to my seminar guide Mrs. Chaya Grover

, Assistant Professor, Department of Electronics & Communication Engineering, for his

valuable suggestions and guidance in the preparation of the seminar report. I express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this seminar successfully.

## ABSTRACT

LabVIEW is a graphical programming environment you can use to quickly and efficiently create applications with professional user interfaces. Millions of engineers and scientists use LabVIEW to develop sophisticated measurement, test, and control system applications using intuitive icons and wires. In addition, the LabVIEW platform is scalable across different targets and OSs. In fact, LabVIEW offers unrivaled integration with thousands of hardware devices and provides hundreds of built-in libraries for advanced analysis and data visualization for you to create virtual instruments you can customize to your needs.

Because LabVIEW programs imitate the appearance and operation of physical instruments, such as oscilloscopes and multimeters, LabVIEW programs are called virtual instruments or, more commonly, VIs. VIs has front panels and block diagrams. The front panel is the user interface. The block diagram is the programming behind the user interface. After you build the front panel, you add code using graphical representations of functions to control the front panel objects. The code on the block diagram is graphical code, also known as G code or block diagram code. In contrast to text-based programming languages, like C++ and Visual Basic, LabVIEW uses icons instead of lines of text to create applications. In text-based programming, instructions determine the order of program execution. LabVIEW uses graphical dataflow programming. In graphical dataflow programming, the flow of data through the nodes on the block diagram determines the execution order. Graphical programming and dataflow execution are the two major reason LabVIEW is different from most other general-purpose programming languages.

## LIST OF FIGURES

**Figure Page No**

Figure 1.1 Type of automation 8

Figure 1.2 plant automation industries 8

Figure 1.3 Numeric Controls and Indicators 15

Figure 1.4. Boolean Controls and Indicators 15

Figure 1.5 string objects are tables and text entry boxes. 15

Figure 1.6. Controls Palette 16

Figure 1.7 Example of a Block Diagram and Corresponding Front Panel. 16

Figure 2.1 Transfer function 17

Figure 2.2 Step response of transfer function. 18

Figure 2.3 Nyquist plot. 18

Figure 2.4 Root locus. 19

Figure 2.5 Tools option 24

Figure.2.6 Boolean of light 25

Figure.2.7 Design 26

Figure 2.8 Vehicle implementation 26

Figure 2.9 Front of project 26

Figure 2.9.1 Front panel of the project. 26

Figure 2.9.2 implementation logic 27

Figure 2.10 Final front panel of project 28

Figure 2.11 Final block diagram of project 28

## Chapter 1

* 1. **Introduction**

LabVIEW is a graphical programming environment that allows you to view every element of your project, including hardware configuration, measurement data, and debugging. This visualization simplifies the integration of measuring gear from any vendor, the representation of complicated logic on the diagram, the development of data analysis algorithms, and the creation of bespoke engineering user interfaces.

LabVIEW applications are referred to as Virtual Instruments or VIs because they replicate the appearance and function of actual instruments such as oscilloscopes and multimeters. Front panels and block diagrams are included in VIs. The user interface is located on the front panel. The programming underlying the user interface is shown by the block diagram.

In contrast to text-based programming languages such as C++ and Visual Basic, LabVIEW creates programs using icons rather than lines of text. Graphical dataflow programming is used in LabVIEW. LabVIEW differs from most other general-purpose programming languages in two main ways: graphical programming and dataflow execution.

Characteristics of LabVIEW:

The following are features of LabVIEW programs:

* + - A graphical and organized appearance
    - Programming based on dataflow and/or events
    - Platform and multi-target capabilities
    - Object-oriented flexibility is a plus.
    - Possibilities for multi-threading

# Industry Automation:

### Automation

Automation, describes wide range of technologies that reduce human intervention in processes. This human intervention is reduced by predetermining decision criteria, sub process relationships and embodying those predeterminations in machines.

Automation includes the use of various control systems for operating equipment such as machinery, processes in factories, boilers. Automation include a higher production rate and increased productivity, more efficient use of materials, better product quality, improved safety, shorter workweeks for labor. We can say that the worker safety is an important reason for automating an industrial operation.

### Robotics Vs Automation

Automation is a process of using physical machines, computer software and other technologies to perform tasks that are usually done by humans, whereas Robotics is the process of designing, creating and using robot to perform a task.

Within industrial automation, robots are used as a flexible way to automate a physical task or process. Collaborative robots are designed to carry out the task in the same way a human

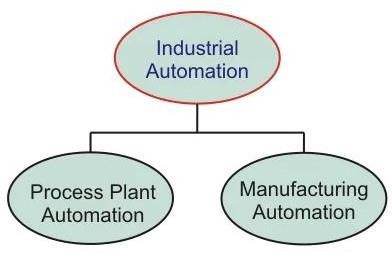
would. More traditional industrial robots tend to carry out the task more efficiently than a human would.

* 1. **Types of Industry Automation:**

Industrial automation is the use of control systems, such as computers or robots, and information technologies for handling different processes and machineries in an industry to replace a human being. It is the second step beyond mechanization in the scope of industrialization.

Types of Industry Automation are:

1. Process Plant Automation
2. Manufacturing Automation



#### Figure 1.1 Type of automation

1. In process plant automation industries, the product results from many chemical processes based on some raw materials. Thus, the overall process plant is automated to produce high quality, more productive, high reliable control of the physical process variables. e.g.: Cement industry, Paper industry, Petrochemical industry, Pharmaceuticals, etc.,

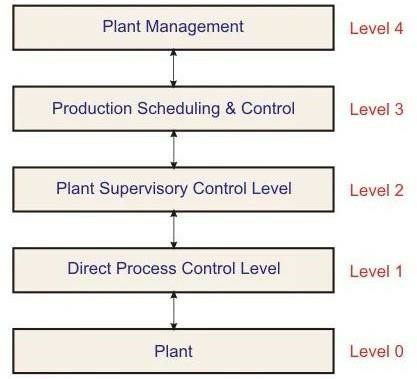


Figure 1.2 plant automation industries

Level 0: plant

Level 1: This involves in the following processes -- data acquisition, plant monitoring, data checking, open and closed loop control, reporting.

Level 2: This involves in the following processes -- plant monitoring performance, optimal process control, plant coordination, failure detection.

Level 3: This involves in the following processes -- production dispatch, inventory control, production supervision, production reporting.

Level 4: This involves in the following processes – market and customer analysis, order and sales statistics, production planning, capacity and order balance.

1. Automation, in the context of manufacturing, is the use of equipment

to automate systems or production processes. The manufacturing industries make the product out of materials using machines/robotics. New trends in manufacturing systems have been using automation systems at every stage such as material handling, machining, assembling, inspection, and packaging. With the computer-aided control and industrial robotic systems, manufacturing automation becomes very flexible and efficient.

e.g.: Some of these manufacturing industries include textile and clothing, glass and ceramic, food and beverages, paper making, etc.

## Benefit of Industry Automation:

Automation of factory or manufacturing or process plant improves production rate through a better control of production. It helps to produce mass production by drastically reducing assembly time per product with a greater production quality. Therefore, for a given labor input it produces a large amount of output.

Integration of various processes in industry with automated machineries, minimizes cycle times and effort and hence the need of human labor gets reduced. Thus, the investment on employees has been saved with automation.

Quality, greater consistency, flexibility, more precise information, safety, cost reduction, improved working conditions.

## Requirement of Industry Automation:

An automation device must be able to handle as many tasks and processes as possible quickly, easily and without errors. It is also important not to forget the human component. That means easy operation, visualization.

It is the lowest level of the automation hierarchy which includes the field devices like sensors and actuators. The main task of these field devices is to transfer the data of processes and machines to the next higher level for monitoring and analysis. And also, it includes the controlling of process parameter through actuators. For instance, we can describe this level as eyes and arms of a particular process.

Sensors convert the real time parameters like temperature, pressure, flow, level, etc. into electrical signals. This sensor data further transferred to the controller so as to monitor and

analyze the real time parameters. Some of the sensors include thermocouple, proximity sensors, RTDs, flow meters, etc.

On other hand actuators converts the electrical signals (from the controllers) into mechanical means to control the processes. Flow control valves, solenoid valves, pneumatic actuators, relays, DC motors and servo motors are the examples of actuators.

# Virtual Instrumentation:

### What is Virtual Instrumentation?

The rapid adoption of the PC in the last 20 years catalyzed a revolution in instrumentation for test, measurement, and automation. One major development resulting from the ubiquity of the PC is the concept of virtual instrumentation, which offers several benefits to engineers and scientists who require increased productivity, accuracy, and performance.

A virtual instrument consists of an industry-standard computer or workstation equipped with powerful application software, cost-effective hardware such as plug-in boards, and driver software, which together perform the functions of traditional instruments. Virtual instruments represent a fundamental shift from traditional hardware-centered instrumentation systems to software-centered systems that exploit the computing power, productivity, display, and connectivity capabilities of popular desktop computers and workstations.

Software that truly provides the leverage to build on this powerful hardware foundation to create virtual instruments, providing better ways to innovate and significantly reduce cost. With virtual instruments, engineers and scientists build measurement and automation systems that suit their needs exactly (user-defined) instead of being limited by traditional fixed-function instruments(vendor-defined).

* + 1. **Virtual Instruments versus Traditional Instruments**

Stand-alone traditional instruments such as oscilloscopes and waveform

generators are very powerful, expensive, and designed to perform one or more specific tasks defined by the vendor. However, the user generally cannot extend or customize them. The knobs and buttons on the instrument, the built-in circuitry, and the functions available to the user, are all specific to the nature of the instrument. In addition, special technology and costly components must be developed to build these instruments, making them very expensive and slow to adapt.

Engineers and scientists whose needs, applications, and requirements change very quickly, need flexibility to create their own solutions. You can adapt a virtual instrument to your particular needs without having to replace the entire device because of the application software installed on the PC and the wide range of available plug-in hardware.

Software is the most important component of a virtual instrument. With the right software tool, engineers and scientists can efficiently create their own applications, by designing and integrating the routines that a particular process requires. They can also create an

appropriate user interface that best suits the purpose of the application and those who will interact with it. With powerful software, you can build intelligence and decision-making capabilities into the instrument so that it adapts when measured signals change inadvertently or when more or less processing sing power is required.

## Benefits of Virtual Instrumentation:

Graphical Programming-One of the most powerful features that LABVIEW offers engineers and scientists is its graphical programming environment. With LabVIEW, you can design custom virtual instruments by creating a graphical user interface on the computer screen.

Flexibility-You can easily add additional functions such as a filter routine or a new data view to a virtual instrument. Costs-Cost reduction is a major advantage of using VI. A computer and the appropriate software are the only things needed for VI, although some virtual instruments may require that several computers be linked together. Occasionally, some hardware may be required. For instance, a video tape deck is sometimes needed to bring certain video formats onto a computer. However, this hardware is almost always cheaper and more efficient than its non-digital counterpart. Other benefits include display, performance, storage etc.

## Requirement of Virtual Instrumentation:

Virtual instrumentation is necessary because it delivers instrumentation with the rapid adaptability required for today’s concept, product, and process design, development, and delivery. Only with virtual instrumentation can engineers and scientists create the user- defined instruments required to keep up with the world’s demands.

The only way to meet these demands is to use test and control architectures that are also software centric. Because virtual instrumentation uses highly productive software, modular I/O and commercial platforms, it is uniquely positioned to keep pace with the required new idea and product development rate.

LabVIEW is an integral part of virtual instrumentation because it provides an easy- to-use application development environment designed specifically with the needs of engineers and scientists in mind. LabVIEW offers powerful features that make is easy to connect to a wide variety of hardware and other software.

## Virtual Programming Platform:

Software is the most important component of a virtual instrument. With the right software tool, engineers and scientists can efficiently create their own applications, by designing and integrating the routines that a particular process requires. They can also create an appropriate user interface that best suits the purpose of the application and those who will interact with it. They can define how and when the application acquires data from the device, how it processes, manipulates and stores the data, and how the results are presented to the user.

With powerful software, you can build intelligence and decision-making capabilities into

the instrument so that it adapts when measured signals change inadvertently or when more or less processing power is required. These subtasks are more manageable and easier to test, given the reduced dependencies that might cause unexpected behavior.

You can design a virtual instrument to solve each of these subtasks, and then join them into a complete system to solve the larger task. The ease with which you can accomplish this division of tasks depends greatly on the underlying architecture of the software.

By employing virtual instrumentation solutions, you can lower capital costs, system development costs, and system maintenance costs, while improving time to market and the quality of your own products.

LabVIEW is an integral part of virtual instrumentation because it provides an easy-to-use application development environment designed specifically with the needs of engineers and scientists in mind. LabVIEW offers powerful features that make is easy to connect to a wide variety of hardware and other software.

One of the most powerful features that [LabVIEW](http://www.ni.com/labview) offers engineers and scientists is its graphical programming environment. With LabVIEW, you can design custom virtual instruments by creating a graphical user interface on the computer screen through which you

* Operate the instrumentation program
* Control selected hardware
* Analyze acquired data
* Display results

## What is LabView

LabVIEW is a graphical programming environment you can use to quickly and efficiently create applications with professional user interfaces. Millions of engineers and scientists use LabVIEW to develop sophisticated measurement, test, and control system applications using intuitive icons and wires. In addition, the LabVIEW platform is scalable across different targets and OSs. In fact, LabVIEW offers unrivaled integration with thousands of hardware devices and provides hundreds of built-in libraries for advanced analysis and data visualization for you to create virtual instruments you can customize to your needs.

Because LabVIEW programs imitate the appearance and operation of physical instruments, such as oscilloscopes and multimeters, LabVIEW programs are called virtual instruments or, more commonly, VIs. VIs has front panels and block diagrams. The front panel is the user interface. The block diagram is the programming behind the user interface. After you build the front panel, you add code using graphical representations of functions to control the front panel objects. The code on the block diagram is graphical code, also known as G code or block diagram code. In contrast to text-based programming languages, like C++ and Visual Basic, LabVIEW uses icons instead of lines of text to create applications. In text-based programming, instructions determine the order of program execution. LabVIEW uses graphical dataflow programming. In graphical dataflow programming, the flow of data through the nodes on the block diagram determines the execution order. Graphical programming and dataflow execution are the two major ways LabVIEW is different from most other general-purpose programming languages.

## LabVIEW Characteristics

LabVIEW programs have the following characteristics:

* A graphical and compiled nature
* Dataflow and/or event-based programming
* Multi-target and platform capabilities
* Object-oriented flexibility
* Multi-threading possibilities

## Graphical and Compiled

While represented graphically, with icons and wires instead of with text, G code on the block diagram contains the same programming concepts found in most traditional languages. For example, G code includes data types, loops, event handling, variables, recursion, and

object-oriented programming. LabVIEW compiles G code directly to machine code so the computer processors can execute it. You do not have to compile G code in a separate step.

## Dataflow and Event-Driven Programming

LabVIEW programs execute according to dataflow programming rules instead of the procedural approach found in most text-based programming languages such as C and C++. Dataflow execution is data-driven, or data-dependent. The flow of data between nodes in the G code determines the execution order. Event-driven programming features extend the LabVIEW dataflow environment to allow the user's direct interaction with the program without the need for polling. Event-based programming also allows other asynchronous activity to influence the execution of G code on the block diagram.

## Multi-Target and Multi-Platform

With LabVIEW applications, you can target multicore processors and other parallel hardware such as field-programmable gate arrays (FPGAs). You can automatically scale LabVIEW applications to CPUs with two, four, or more cores, often with no additional programming effort. G code, with the exception of a few platform-specific functions, is portable between the different LabVIEW systems for different operating systems. Therefore, you can often use the same code whether running LabVIEW on Windows, Mac OS X or Linux systems.

## Object-Oriented

Object-oriented programming is a popular programming approach across a wide variety of programming languages. It allows a variety of similar, yet different items, to be represented as a class of objects in software. LabVIEW provides tools and functions so you can use object- oriented programming techniques in G code.

## Multithreading and Memory Management

LabVIEW enables your code to have automatic parallelism. In other languages if you want to run code in parallel, you have to manage multiple threads manually. The LabVIEW environment, with the compiler and execution system working together, automatically runs code in parallel whenever possible. Most of the time the details of the execution system are unimportant to you because the system does the right thing without intervention. However, LabVIEW also provides you with options for improving performance.

## Features of LabView

### Controls and Indicators

LabView comes with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates. Figure 1-7 has the following objects: two controls: Number of Measurements and Delay(sec). It has one indicator: an XY graph named Temperature Graph. The user can change the input value for the Number of Measurements and Delay(sec) controls. The user can see the value generated by the VI on the Temperature Graph indicator. The VI generates the values for the indicators based on the code created on the block diagram. Every control or indicator has a data type associated with it. For example, the Delay(sec) horizontal slide is a numeric data type. The most commonly used data types are numeric, Boolean value and string.

### Numeric Controls and Indicators

The numeric data type can represent numbers of various types, such as integer or real. The two common numeric objects are the numeric control and the numeric indicator, as shown in Figure 1.3 Objects such as meters and dials also represent numeric data.

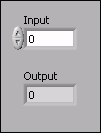


Figure 1.3 Numeric Controls and Indicators

### Boolean Controls and Indicators

The Boolean data type represents data that only has two possible states, such as TRUE and FALSE or ON and OFF. Use Boolean controls and indicators to enter and display Boolean values. Boolean objects simulate switches, push buttons, and LEDs. The vertical toggle switch and the round LED Boolean objects are shown in Figure 1.4.

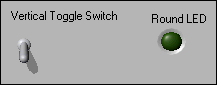


Figure 1.4. Boolean Controls and Indicators

### String Controls and Indicators

The string data type is a sequence of ASCII characters. Use string controls to receive text from the user such as a password or user name. Use string indicators to display text to the user. The most common string objects are tables and text entry boxes as shown in Figure 1.5.

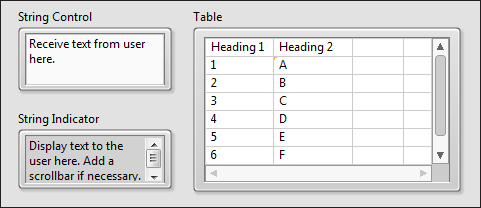
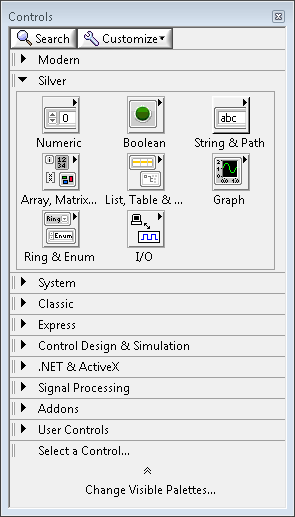


Figure 1.5 string objects are tables and text entry boxes.

### Controls Palette

The Controls palette contains the controls and indicators you use to create the front panel. You access the Controls palette from the front panel window by selecting View» Controls Palette. The Controls palette is broken into various categories; you can expose some or all of these categories to suit your needs. Figure 1-11 shows a Controls palette with all of the categories exposed and the Silver category expanded. During this course, most exercises use controls from the Silver category.



### Block Diagram

Figure 1.6. Controls Palette

Block diagram objects include terminals, subVIs, functions, constants, structures, and wires, which transfer data among other block diagram objects.

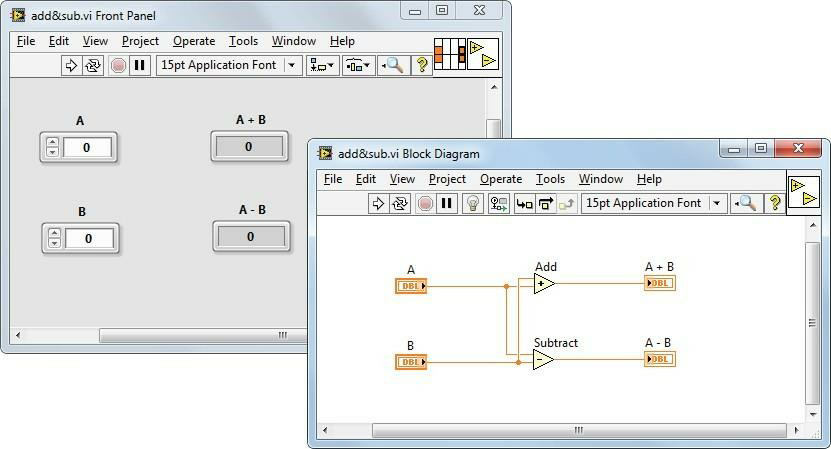


Figure 1.7 Example of a Block Diagram and Corresponding Front Panel.

## Chapter-2

* 1. **What I have learned**

I have learned the basic knowledge require to use LabVIEW and perform operation on it. Basic knowledge of the structure of the LabVIEW option available to use. Data type president in the LabVIEW that include Numeric, arrays, string, Boolean, loop, function. using these base functions, I performed given task some the task is

* + - generating transfer function,
    - step response of the transfer function,
    - root locus,
    - Nyquist plot,
    - Dice simulation,
    - window media player using LABVIEW
    - Fan control using LabVIEW
    - Generating transfer function

## Transfer function

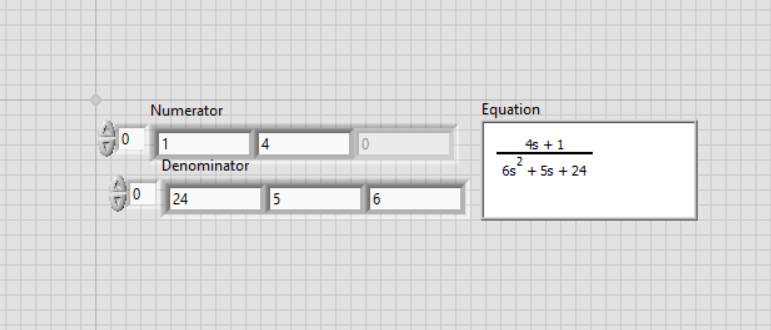


Figure 2.1 Transfer function

## Step response of the transfer function

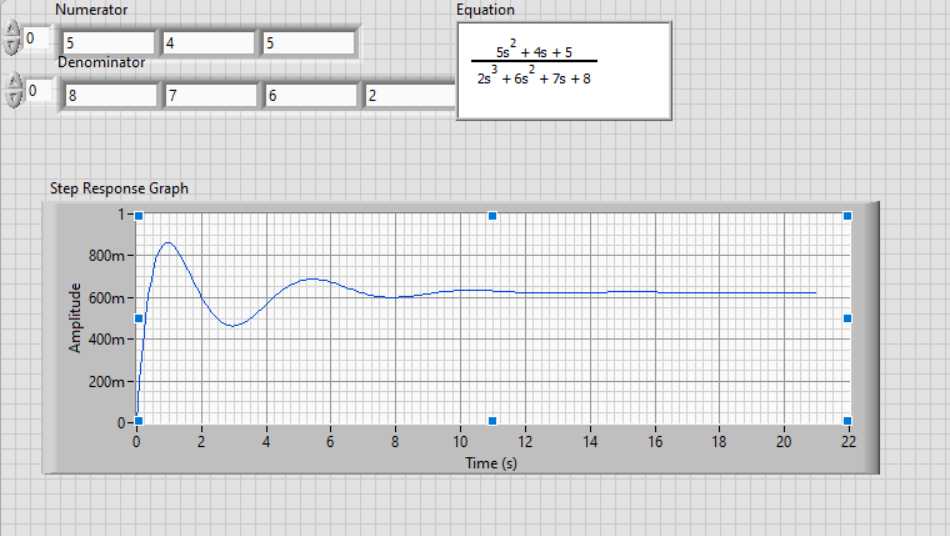
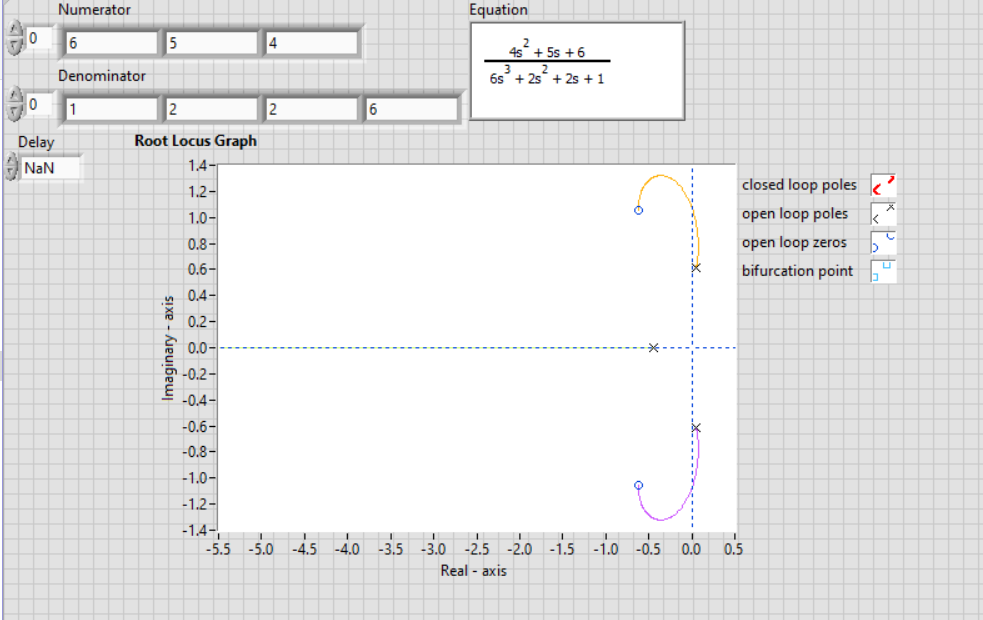


Figure 2.2 Step response of transfer function.

### Root Locus



**2.1.3 Nyquist plot**

Figure 2.3 Root locus.

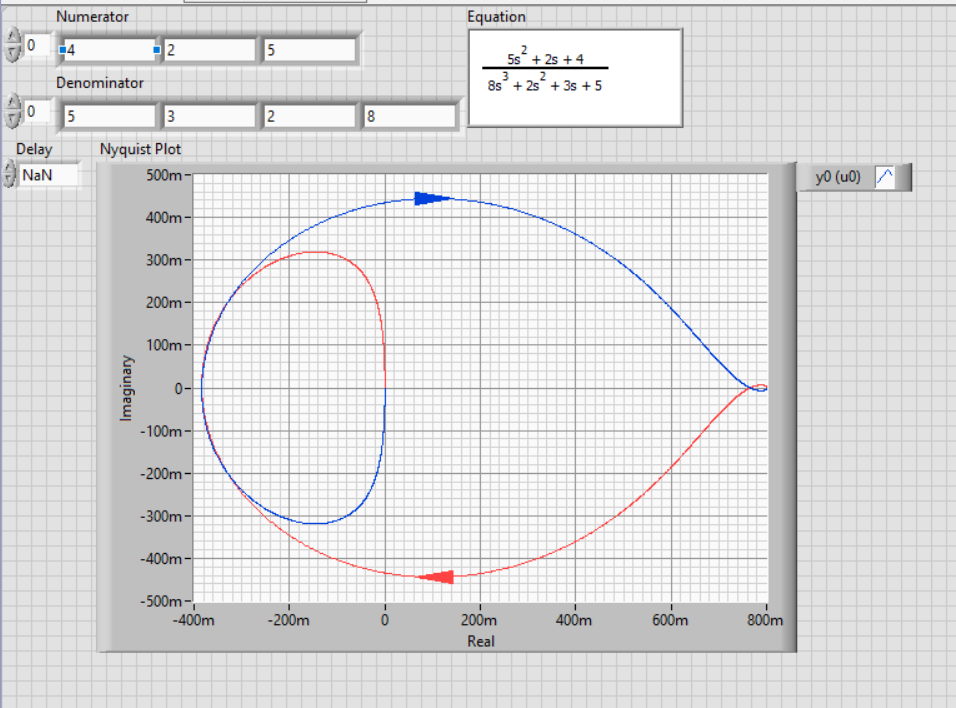


Figure 2.4 Root locus.

## APPLICATION of LABVIEW

* + 1. **Speech signal processing**

The speech recognition and processing are playing a vital role in various industries including data acquisition, hands-free mobile manufacture for physically challenged, instrumentation field and machine control. For implementing the speech recognition system in LabVIEW platform, the prerequisites include sound card, microphone and other accessories related to the hardware PC/Mobile. In the design of intelligent voice system using LabVIEW, Virtual Instrument (VI) can be incorporated to simplify the process of Voice Recognition Technology (VRT). It has been enriched with the features of graphical-based programming, digital data stream and simplified programming model to involve the audio front panel. Even though speech signal of different people may have similar power spectrum, the low frequency band will differ uniquely for each and every user. Therefore, after recording the voice it can be analyzed with the help of sound acquisition module in LabVIEW. Once speech signal is received from the microphone, it is preprocessed and filtered before adding the Hamming window to calculate the power spectrum of each user. From the power spectrum, low frequency peaks can be obtained in order to identify the voice characteristics of individual user. Using LabVIEW PSD module peak frequency is extracted from the spectrum of voice signal.

## LabVIEW for embedded systems

LabVIEW, an easier graphical programming environment than GUI based application development platforms like Visual Basic and Visual C++. The conventional instrumentation does not perform analysis and controlling of multiple signals at the same time but LabVIEW with Virtual Instrumentation will do so. The main advantage of LabVIEW is that it can be used for PC-based monitoring and data logging.

The latest processors that have been used for the embedded system design are ARM microcontrollers. LabVIEW has a comprehensive graphical development environment module for ARM microcontrollers. This has been jointly developed by Keil - An ARM Company and NI instruments for Embedded System design applications. This module will reduce the development cost and fasten the programming part for Embedded System Design. This module includes hundreds of analysis and signal processing functions, integrated I/O, and an interactive debugging interface. These advantages of LabVIEW pay a path for its usage in embedded applications for controlling. There are many case studies for such kinds of applications.

## Applications of LabVIEW in robotics and VLSI

Present world is looking for high throughput fast processing system with features that meets ideal specifications. As we know, a system complexity increases consequently the system hardware and software must be complicated to attain the desired target within the time specification. As we are going through the era of artificial intelligence and machine learning so that it is necessary to add technological improvements in our system designs to compete with latest market trends. In the field of robotics, software and hardware have equal responsibility and functionality to attain the accuracy while take part in a certain action. In order to create a stable perfect system, the designer can choose graphical based design rather than conventional system design.

LabVIEW can easily use as a flexible environment for the robotic design. LabVIEW is a development environment that will provide flexibility for automated product design validation and monitoring of a machine status. Here the designer can use the language “G” for the system coding and drag and drop option of appropriate functional blocks will generate a working system according to the specifications.

Industrial machine builders are forced to deliver machines that meets several demands. It is necessary to hold their unique position in the respective field with respect to the competitors. LabVIEW is mainly used for building a graphical user interfaces for systems and also it is utilizing inherent parallelism for instrumentation, control system and robotics. If the designers are looking for a simple design in the hardware and software LabVIEW is the best option for that. Robotic arm control mechanism needs more accuracy to work in the fields such as of medical electronics and chemical industry. The processor will run according to the program coded using lab view and the processor will generate the control signals to control and monitor the movement of robotic arm.

LabVIEW is applicable for standalone as well as moving system design. Wherever the application of a mixed system comes then the design complexity will increase and monitoring will be more complicated. Here the graphical user interface will be more useful to control the action of the system and its programming.

## Medical applications

Jung et al, integrated the flexible multimodal sensor-based E-Skin which will mimic the skin in human beings. The sensors like temperature, pressure, flow and humidity everything has been included to create the characteristics of human skin. To measure the varying resistance in the hair sensor they customized the PC- Controlled Servomotor LabVIEW Tower Pro SG 90 and PC-recordable acquisition board NI USB-6218. The E-Skin plays vital role in soft electronics to enhance the Human-Machine Interface (HMI) for analyzing the ECG, EMG and other parameters. The wearable and flexible electronics [[13](https://www.intechopen.com/chapters/75665#B13)] paves the way for comfort in using these devices.

## Education

Many real time remote –based laboratory models have been constructed to help the students in schools and higher education institutions to perform the research and gain the knowledge from their own location itself. Several laboratories including the light sensing experiment designed by Singh et al. using LabVIEW, LDR and LM35 sensor, the phasor estimation algorithm using NI ELVIS kit designed by Mondal et al, operational process monitoring application using SHRIMP VL application, analog electronics experiment using “Red Pitaya” etc. The LabVIEW has been effectively and efficiently included in many studies in literature to enhance and boost the research community. Polat et al, designed the Distance Vocation Education and Training D-VET to monitor the lighting levels of LED lamps in remote laboratory with the help of PID controller, Sensor, Actuator, NI-IMAQ software and Compatriot DAQ.

Azenova et al. designed an e-learning course for second year industrial students under the category of laboratory experiments in electronics. The LabVIEW is incorporated in the Goodyear e-learning model they have adopted for implementing to their students. The basic laboratory experiments like clipper, clamper, inverting and non-inverting amplifier, summing and differential amplifier were simulated with the help of virtual instruments front panel in LabVIEW.

## Miscellaneous applications

The fault detection in analog circuit can be easily analyzed with the help of LabVIEW interfaces using deep forest learning method. Zhen et al. have designed the fault diagnosis scheme with the Waveform generator Agilent 33250, Data acquisition module NI 1042q, LabVIEW, Digital oscilloscope Agilent 54853 along with power supply and testing circuit. The outputs of testing circuit are collected by data acquisition module and stored in the LabVIEW and used for further diagnosis.

For assessing the quality of water to ensure the living of aquatic species like fishes in a healthy environment. Othman et al, proposed a system using the sensors to measure the pH level and temperature of the water in the tank. The data collected from the sensors are given to the Data Acquisition System (DAQ) NI myRIO-1900. Then the data will be analyzed periodically in the workstation to ensure the safety of aquatic animals. The interface between DAQ and workstation is created with the help of LabVIEW software. The temperature and pH measurements are displayed in the front panel of LabVIEW.

Ahmed et al. discussed the implementation of LabVIEW software to control the FESTO MPS PA compact workstations as part of remote laboratory. In the paper they designed the Easyport USB data acquisition card based on activeX elements to interface with LabVIEW and control the applications in Fluid Lab. To obtain the real time information from the Sensors and actuators of MPS-PA station various VI were included. To incorporate remote monitoring web server of LabVIEW is required. With the help of the VI the user can adjust the parameters of Controller and with help of IP camera and web server the MPS PA compact workstation can be monitored remotely. Proportional-Integral-Derivative (PID) and ON/OFF controllers these two experiments as part of laboratory course were implemented using the system designed by Ahmed et al.

Mikhailov et al. aimed to monitor the specific content from the broadcast channels using the help of Cognitive Radio (CR). For the real time data analytics, particular subset of channels from the selective frequency range will be chosen and the significant information in the subset will be relayed for monitoring in the remotely using the VI from LabVIEW. For monitoring Software Defined Radio (SDR) technology USRP 2900/2901 and LabVIEW were utilized in the design. The key factor in the application is to either receive & process or transmit the subset of signals in the digitized format with the help of the packages available in the LabVIEW.

## Project

### Project Aim and Intro

Aim: Design of two pole traffic light system using LabVIEW and DSC with traffic light.

ABSTRACT: As the problem of urban traffic congestion spreads & occurrence of road accidents increase, there is a pressing need for the introduction of advanced technology and equipment to improve the traffic control algorithms to better accommodate this increasing demand. The simplest way for controlling a traffic light is using timer for each phase.

Another way is to use electronic sensors in order to detect vehicles, and produce signal that cycles. In this paper we propose the LABVIEW simulation model for controlling the traffic lights based on time interval. This simulation model can extend to control the time interval of the traffic light based on traffic density system for controlling the traffic light by any other suitable method

INTRODUCTION: According to the World Health Organization report, India records the highest number of road accidents deaths per year. About 5 lakh accidents take place on Indian roads killing about 1.3 lakh people and injuring about 5.2 lakh each year. These

numbers translate one accident every minute and one death in road accidents every four minutes which is an alarmingly high number.

Traffic problems now-a-days are increasing because of the growing number of vehicles and the limited resources provided by current infrastructures. So, a simulating and optimizing traffic control algorithms for increasing demand is needed of the time. The simplest way for controlling a traffic light is to use a timer for each phase and the LabVIEW Simulation model for controlling the traffic lights based on time interval. This Simulation model can be extended to control the time interval of the traffic light based on traffic density. This can be even extended to integrated traffic management system for a metropolitan city based on the density of traffic.

Automatic traffic light is controlled by timers and electrical sensors. In classical traffic light system, each phase has a constant numerical value loaded in the timer. The lights are automatically getting ON and OFF depending on the timer value changes. When properly used, traffic control signals are important devices for the control of vehicular in road. They assign the right of-way to a choice of traffic movements and there by deeply influence traffic flow. Traffic control signals that are properly designed, located, operated, and maintained will have one or more of the following advantages:

1. Provide orderly movement of traffic
2. Minimize completing movement
3. Coordinated for continuous movement
4. Provide driver confidence by assigning right way.

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine the order of program execution, LabVIEW uses dataflow programming, where the flow of data through the nodes on the block diagram determines the execution order of the VIs and functions.

VIs, or virtual instruments, are LabVIEW programs that imitate physical instruments. Though there are conventional methods that are still useful and relevant, LabVIEW based Traffic control system is relatively easier approach in operators’ point of view who operate the traffic control system because it is very easier to design, redesign, debug in LabVIEW as it is a Graphical Programming language.

They have predefined cyclic time which schedules off-line on a central computer based on average traffic conditions. Present Traffic Light Controllers (TLC) are based on microcontroller and microprocessor. These TLC have limitations because it uses the pre- defined hardware, which is functioning according to the program that does not have the flexibility of modification on real time basis.

It is possible to design such system to overcome daily problems of traffic congestion using graphical programming language Lab VIEW. Among a variety of general-purpose programming platforms National Instrument’s LabVIEW is widely used graphical code development environment which allows system level developers to perform rapid prototyping and testing. It is supported by a powerful and rich collection of pre-written library functions and programming tools meant to accomplish various tasks related to user- controlled applications for equipment interface, laboratory measurements, data visualization and analysis.

## 2.32 Project Description

In this section, we concentrate on the design of the traffic light using LabVIEW. The LabVIEW template VIs include the sub-VIs, functions, structures, and front panel objects you need to get started building common measurement applications. LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters VI contain the following three components:

* + - * Front panel —Serves as the user interface.
      * Block diagram — Contains the graphical source code that defines the function of the VI.
      * Icon and connector pane — Identifies the VI so that you can use the VI in another VI. A VI within another VI is called a sub VI.
        + First open LabVIEW and create a new vi. Go to the block diagram and from tools -> DSC module -> image navigator.
        + First open LabVIEW and create a new vi. Go to the block diagram and from tools -> DSC module -> image navigator.

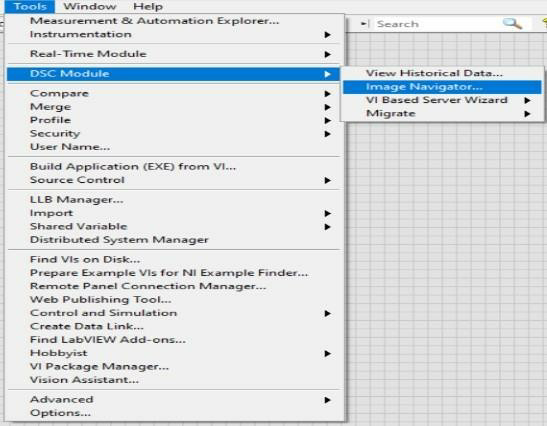


Figure 2.5 Tools option

* + - * + From image navigator select the road from conveyor belts for the roads and set them on four side. Now from controls select round LEDs requires for the traffic lights.

By advanced settings select the customized option for the lights. The lights are customized from the DSC module same as the roads taken.

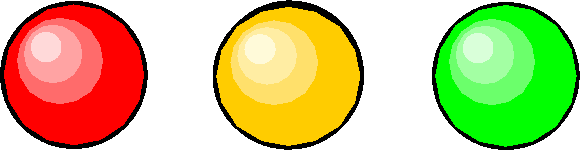


Figure.2.6 Boolean of light



Now arrange the lights in the form of vertical by using decorations. The lights will be in the order of Red, Yellow and Green. These light sets will be two sets named as pole 1 and pole 2.

* Arrange the two poles one the sides of the two roads and taking the vertical and horizontal pointer slides place them on the roads.

The arrangement is shown below.

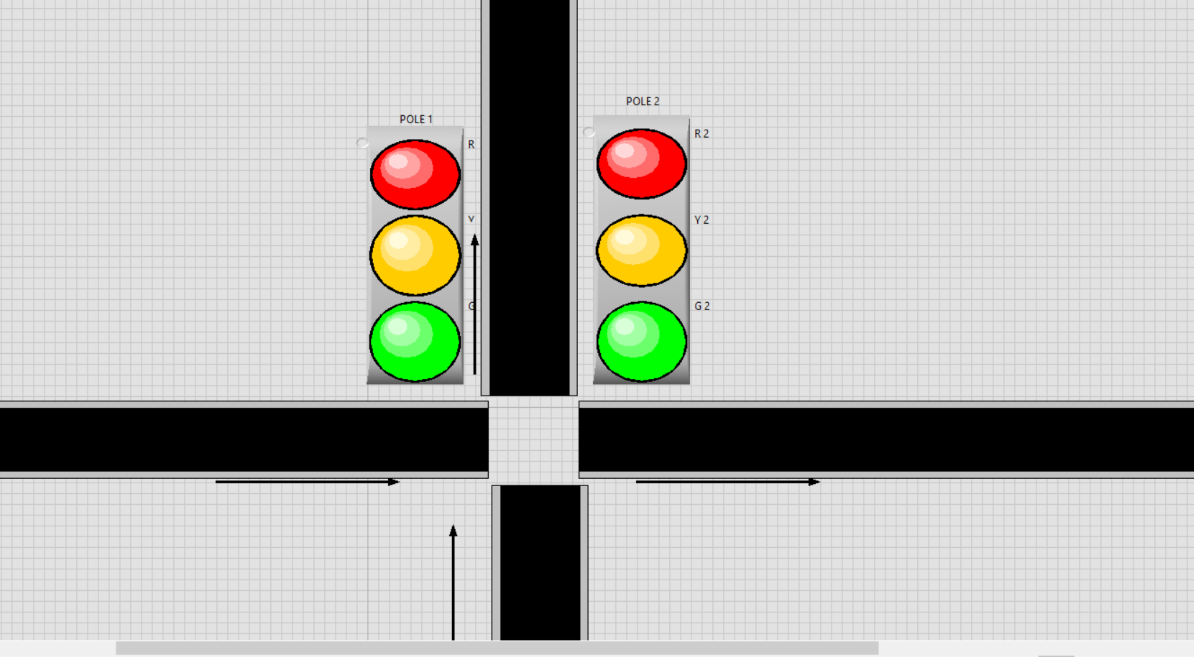


Figure.2.7 Design

* From the DSC module take the vehicles of your choice and set them on the pointer slides. The vehicles will be on each road.
* The figure below will show the total arrangement of the traffic system.

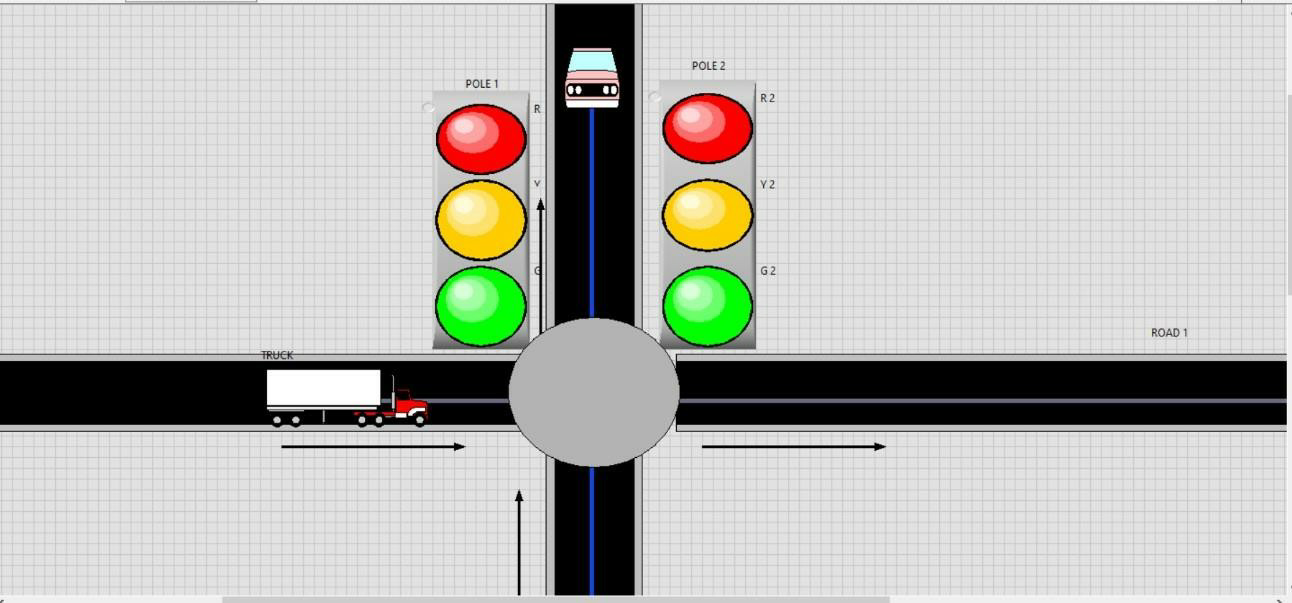


Figure 2.8 Vehicle implementation

* Now coming to the front panel, the logic for the traffic light system is designed.
* Take a while loop and case structure and arrange the Booleans in the order, create constant for every Boolean.

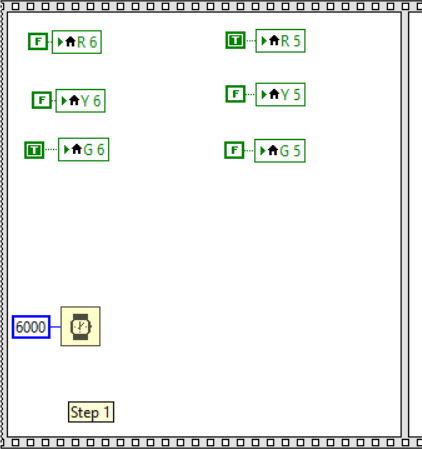


Figure 2.9 Block diagram of project

* From the above diagram it shows the logic for the red light. In the same way by adding the case after write the logic for the yellow and green lights. And the required timing for the next light to be lighten.
* The cases are shown as

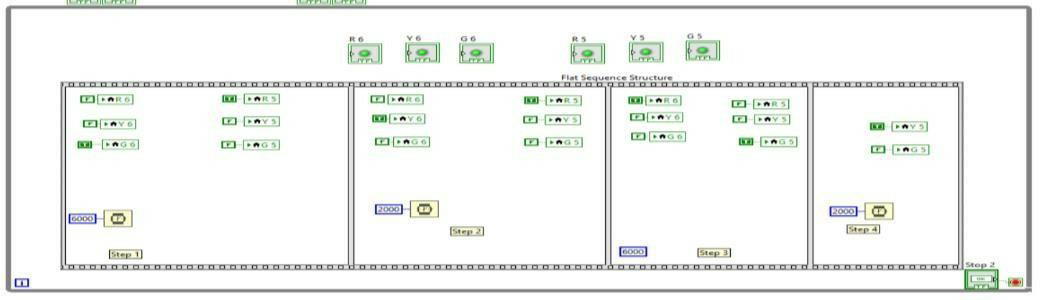


Figure 2.9.1 block diagram of the project.

* For the vehicles to move the logic should be written. For the logic take a case structure and while loop for the truck and the timing to move. The logic is show below. The logic should be written in step 1 i.e., first frame.

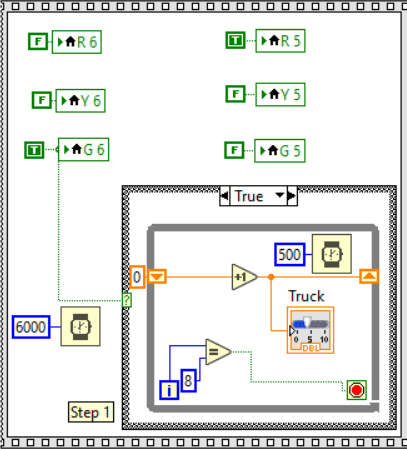


Figure 2.9.2 implementation logic.

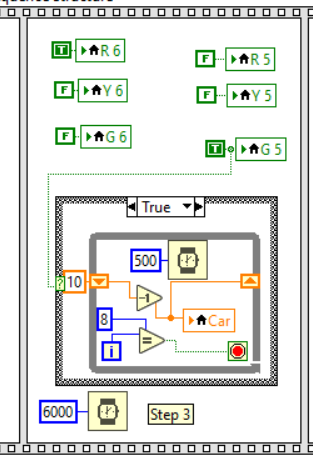
* As the same in figure.7 the logic should be written for the car and the logic is shown as in figure.8

Figure 2.9.3 Front panel of the project.

* After the completion of the total design run the program to move the car and truck and the function of the traffic lights.

## 7.3 Front panel and Block Diagram

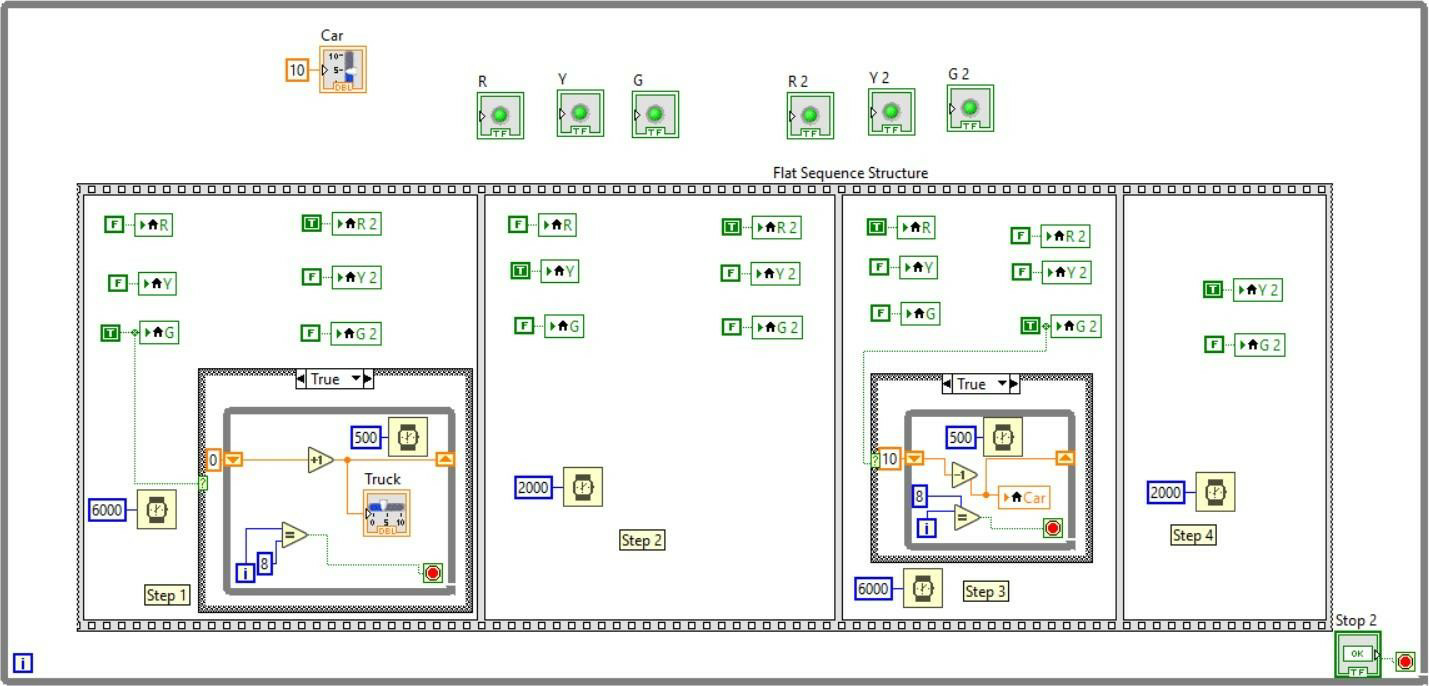
Front panel:

Figure 2.10 Final front panel of project

Block Diagram:

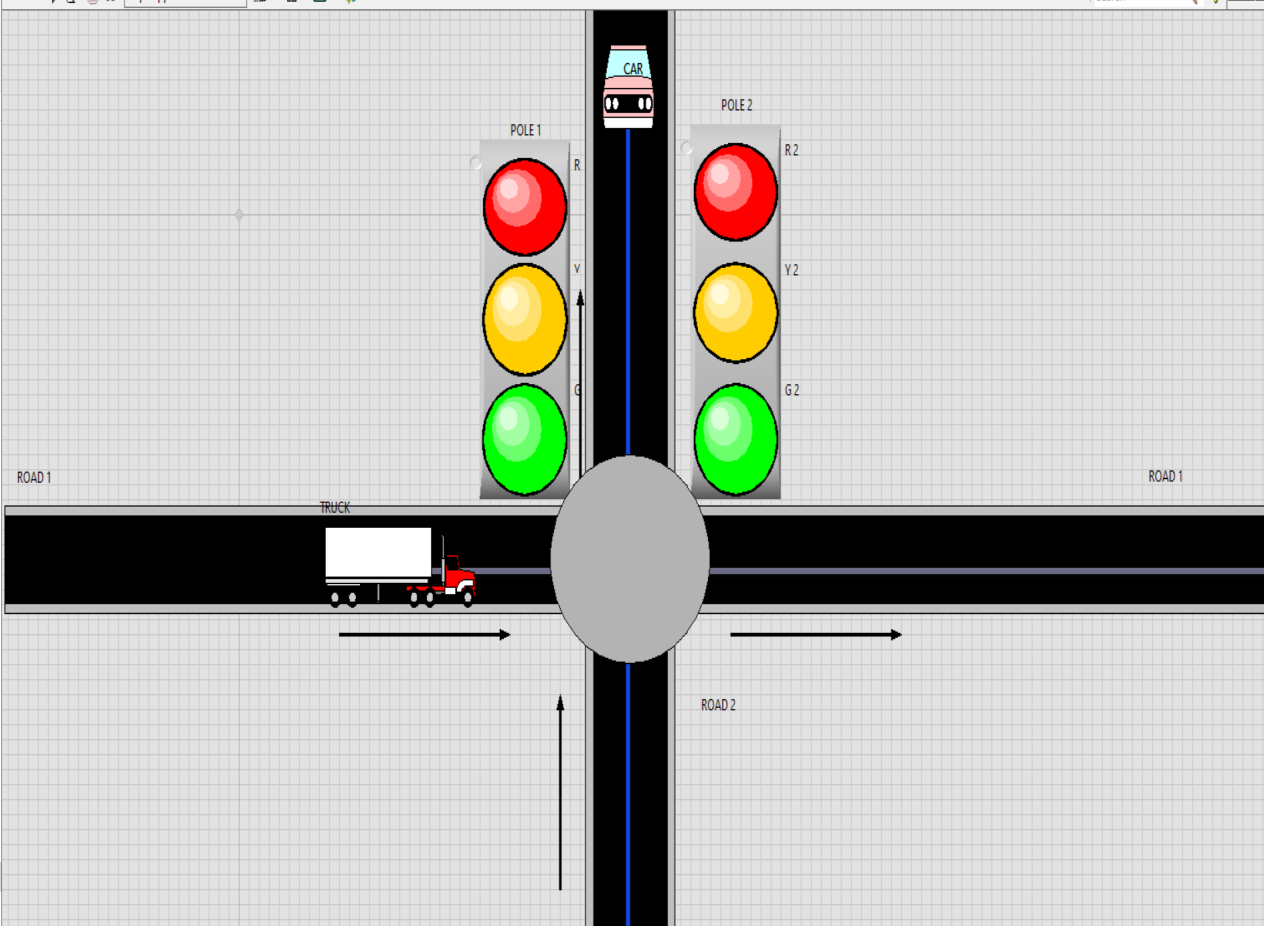


Figure 2.11 Final block diagram of project

* 1. **Outcome:**

An automatic Traffic control system is very important for traffic management in rapidly growing metropolitans and cosmopolitans. Though there are conventional methods that are still useful and relevant, LabVIEW based Traffic control system is relatively easier approach in operators’ point of view who operate the traffic control system because it is very easier to design, redesign, debug in LabVIEW as it is a

Graphical Programming language. This model can also be extended to program the timers depending on density. Also, this project can be extended to design Volume 2, Issue 6, November-December-2017 | [www.ijsrcseit.com |](http://www.ijsrcseit.com/) UGC Approved Journal [ Journal No:64718] 542Traffic control using an image processing technique. The Design and implementation of a LabVIEW based Integrated Traffic Management system would be very useful and successful

### REFERENCES

* + - YouTube <https://www.youtube.com/>
    - Wikipedia <https://www.wikipedia.org/>
    - National instruments <https://www.ni.com/en-in.html>
    - Research Gate <https://www.researchgate.net/>