LABORATORY MANUAL

ELECTRONICS DESIGN LAB

Department of Instrumentation Engineering

JORHAT ENGINEERING COLLEGE

Assam-785007

DOs

1. Log of the system properly before switching off

2. Be punctual, maintain discipline & silence

3. Leave your shoes in the rack outside

4. Handel the equipments carefully

5. save all yours file properly

6. Report any problem to the computer with the person in charge.

Don’ts

1. Avoid unnecessary chat or walk

2. Avoid stepping on electrical wires or any other computer cables.

3. Avoid using cell phones unless absolutely necessary

4. Do not insert metal objects such as clips, pins and needles into the computer casings. They may cause fire.

5. Do not insert personal pen drive& don’t remove anything from the computer laboratory without permission.

6. Do not touch, connect or disconnect any plug or cable without your lecturer/laboratory technician’s permission.

7. Avoid late submission of laboratory reports.

6THSEMESTER BE (IN)

Electronics Design Lab (EE181614)

COs	Course Outcomes
CO1	Design electronic systems to meet the requirements of society, academia and industry.
CO2	Analyze the performance of electronic system after completion of its design.
CO3	Work as individual and group.
CO4	Adopt professional ethics and responsibilities.
CO5	Communicate instruction delivery for working and writing reports.

Mapping of Course outcome and Programme outcome and Programme Specific Outcome:

Course Outcome	Programme Outcome												PS01	PS02
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		
CO1	2	2	1	1	3	2	1					1		2
CO2	2	2	1	1	3							1		
CO3								2	2	2		1		
CO4								2	2			1		
CO5								1	1	2		1		

EE181614	Electronics Design Lab	**Semester-VI**		

Experiment No	Title of the experiments	Objective of the experiments
1	Designing of temperature converter	To design a Fahrenheit to Celsius converter on LabVIEW & display.
2	Design of water level indicator	To develop a water level indicator using LabView and display the output.
3	Designing of traffic light signal	To develop a traffic signal using LabView
4	Designing of timer circuit	To develop a simple timer in the LabVIEW and display the time count in LabVIEW
5	Interfacing of stepper motor by using 8085 microprocessors	Interfacing/control of stepper motor motion in clockwise & counter clockwise direction by using 8085 microprocessors

INTRODUCTION TO LABVIEW

National instruments Labview is a graphical programming language that has its root in automation control and data acquisition. Its graphical representation, similar to process flow diagram, was created to provide an initiative programming environment for scientist and engineers. The language has matured over last 20 years to become a general purpose programming environment. Labview has several key features which makes it a good choice in an automation environment. These include simple network communication, turnkey implementation of common communication protocols (RS232,GPIB,etc),powerful toolsets for process control and data fitting, fast and easy user interface construction, and an efficient code execution environment.

Ni LABVIEW background

LabVIEW, which stands for Laboratory Virtual Instrumentation Engineering Workbench is a graphical programming language first released in 1986 by National Instruments (Austin, TX). LabVIEW implements a dataflow paradigm in which the code is not written, but rather drawn or represented graphically similar to a flowchart diagram.

Program execution follows connector wires linking processing nodes together. Each function or routine is stored as a virtual instrument (V) having three main components: the front panel which is essentially a form containing inputs and controls and can be displayed at run time, a block diagram where the code is edited and represented graphically, and a connector pane which serves as an interface to the VI when it is imbedded as a sub-VI.

Unlike most programming languages, LabVIEW compiles code as it is created thereby providing immediate syntactic and semantic feedback and reducing the time required for development and testing. Writing code is as simple as dragging and dropping functions or Vis from a functions palette onto the block diagram withín process structures (such as For Loops, or Case Structures) and wiring terminals (passing input values, or references). Unit testing is simplified because each function is separately encapsulated; input values can be set directly on the front panel without having to test the containing module or create a separate test harness. Memory allocation/deallocation is automatically managed by LabVIEW.

Similar to most programming languages, LabVIEW supports all common data types such as integers, floats, strings, and clusters (structures) and can readily interface with external libraries, ActiveX components, and .NET framework.

The latest version of Labview are labview 2021 (released in August 2021) and LabVIEW. NXG 5.1 (released in JANUARY 2021).NI release the free for non-commercial use LabVIEW and LabVIEW NXG community edition on April28,2021

EXPERIMENT NO: 01

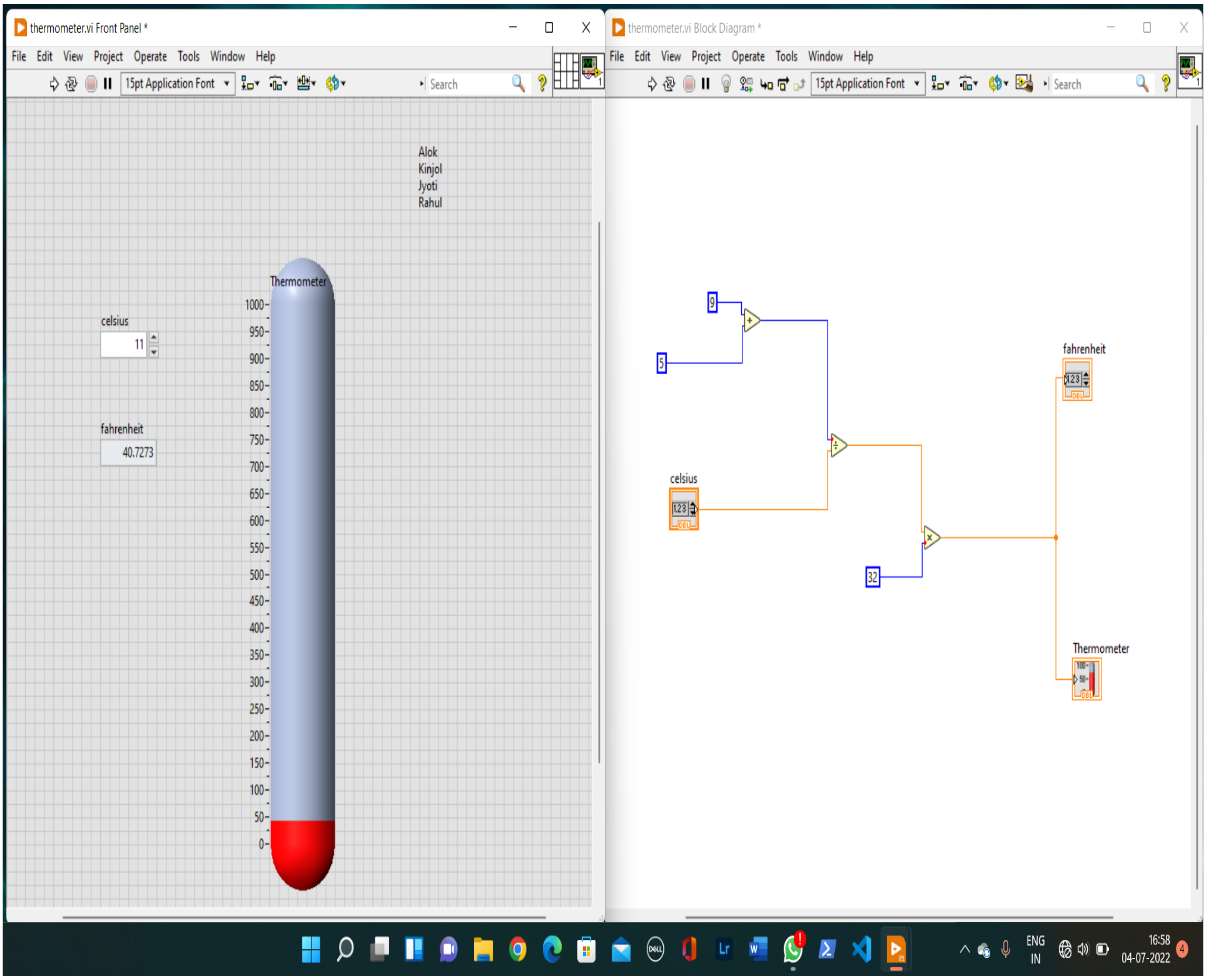
AIM OF THE EXPERIMENT: Designing of temperature converter.

OBJECTIVE: To design a Fahrenheit to Celsius converter on LabVIEW & display.

SOFTWARE USED: LABVIEW 2019

LABVIEW COMPONENTS: Thermometer, numeric

BLOCK DIAGRAM:



PROCEDURE:

1. Go to the Front Panel and right click on it.

2. Now, go to the Controls -> Modern -> Numeric -> Numeric Control.

3. Select this block and placed it on the front panel.

4. Changed in name to Celsius.

5. Now, go to Controls -> Modern -> Numeric -> Numeric Indicator.

6. Selected this block and place it on the Front Panel

7. Named it as a Fahrenheit.

8. Select another Numeric Control and place it similarly on the Front Panel and name it Kelvin.

9. Go to Functions -> Express -> Arithmetic & Comparison -> Formula.

10. Select this block and place it on the "Front Panel" window.

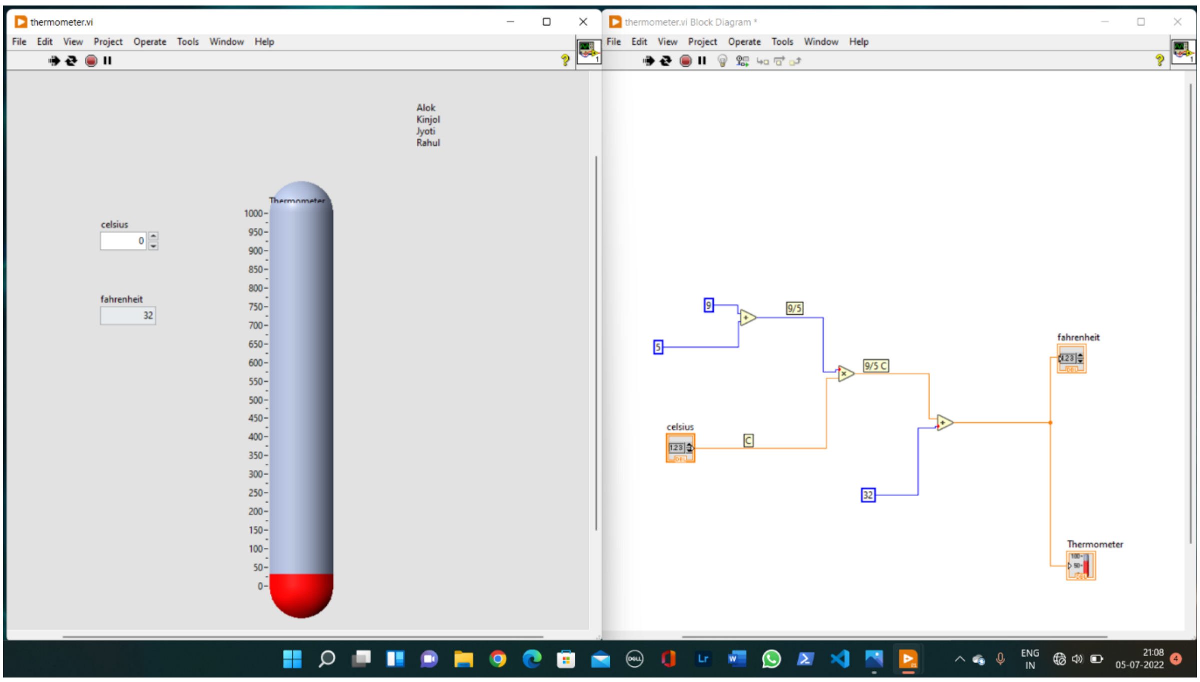
11. Now, I have changed the formula in order to convert "Celsius" to "Fahrenheit".

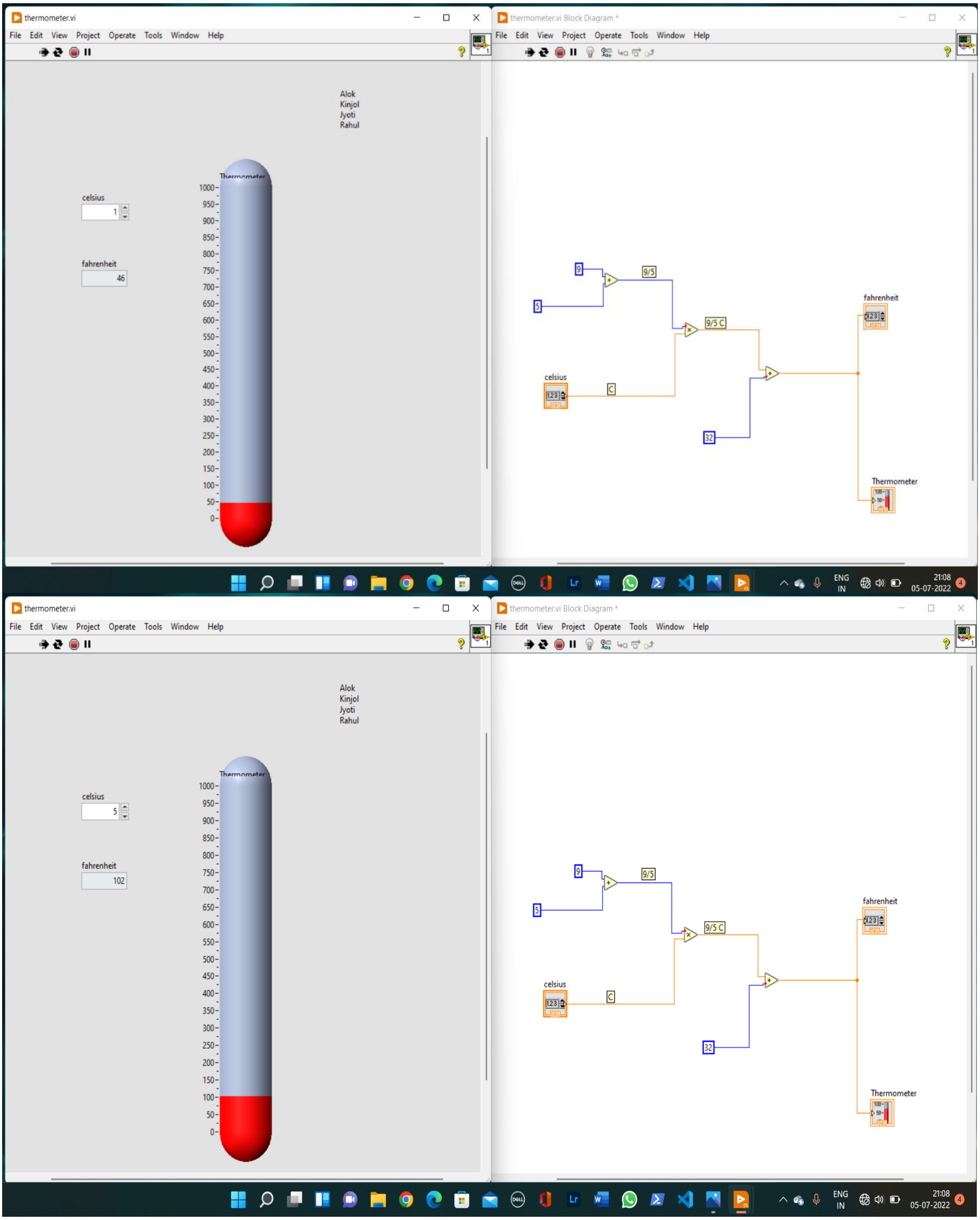
12. After changing the formula pressed "OK".

13. Now, connected the "Celsius" with the "X1" terminal of the "Formula" block and connect "Result" with the "Fahrenheit" block.

14.Now, run the program.

RESULT





CONCLUSION: When we run the program, with the change of the value of Celsius, the value of Fahrenheit also vary and the thermometer shows the variation of temperature.

EXPERIMENT NO-02

AIM OF THE EXPERIMENT- Design of Water level Indicator

OBJECTIVE- To develop a water level indicator using LabView and display the output.

Software- LabView 2019

LabVIEW programs are represented by files called VIs, which stands for Virtual Instrument. A VI may be the main VI that starts a program, or it may be a sub-VI that is called by another VI. Regardless, all VIs has two parts: a Front Panel and a Block Diagram

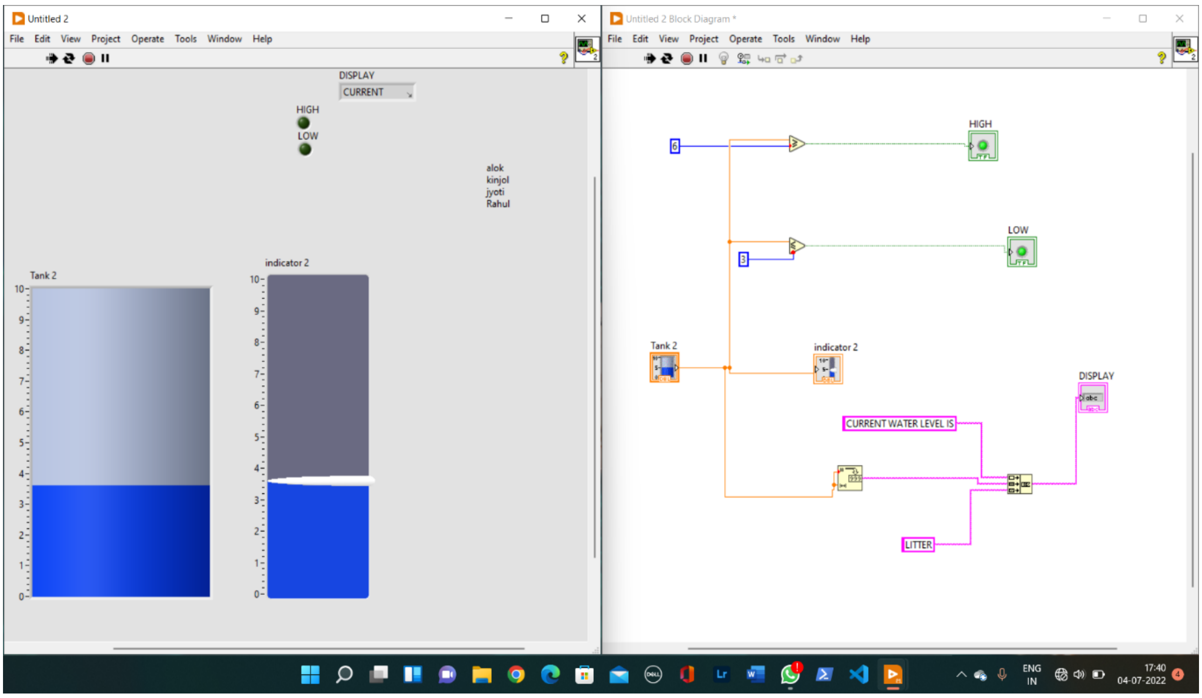
Front Panel- This is the window that users will see when they run the program. It includes controls where the user operates the program or change its parameters. It also includes indicators to display the outputs from the VI. If users might actually view this front panel, then controls and indicators can be graphical in nature. For the sub-VIs where users will not typically view the front panel, the controls and indicators are usually simple values.

Block Diagram- The logic of the VI is shown in this window. Icons on the diagram describe operations to perform on the data. Wires show how the data flows from one icon on the screen to another. Data flows from the controls on left to the indicators on the right.

LabView Components-

1.Tank 2.Vertical pointer 3. LED for high- and low-level indication 4. Display

BLOCK DIAGRAM



1. Go to the “Front Panel” and press “Right Click” from your personal computer or laptop.

2. Now, go to the Controls-> Modern-> Numeric-> Tank.

3. Now go to the “Front Panel” and then go to the Controls-> Modern-> Numeric-> Vertical Pointer.

3. We changed the name of vertical slider as indicator.

4. we take two Boolean LEDs and changed the name as high and low.

5. we take a numeric indicator to display the level of water.

6. we connect the water tank to the indicator.

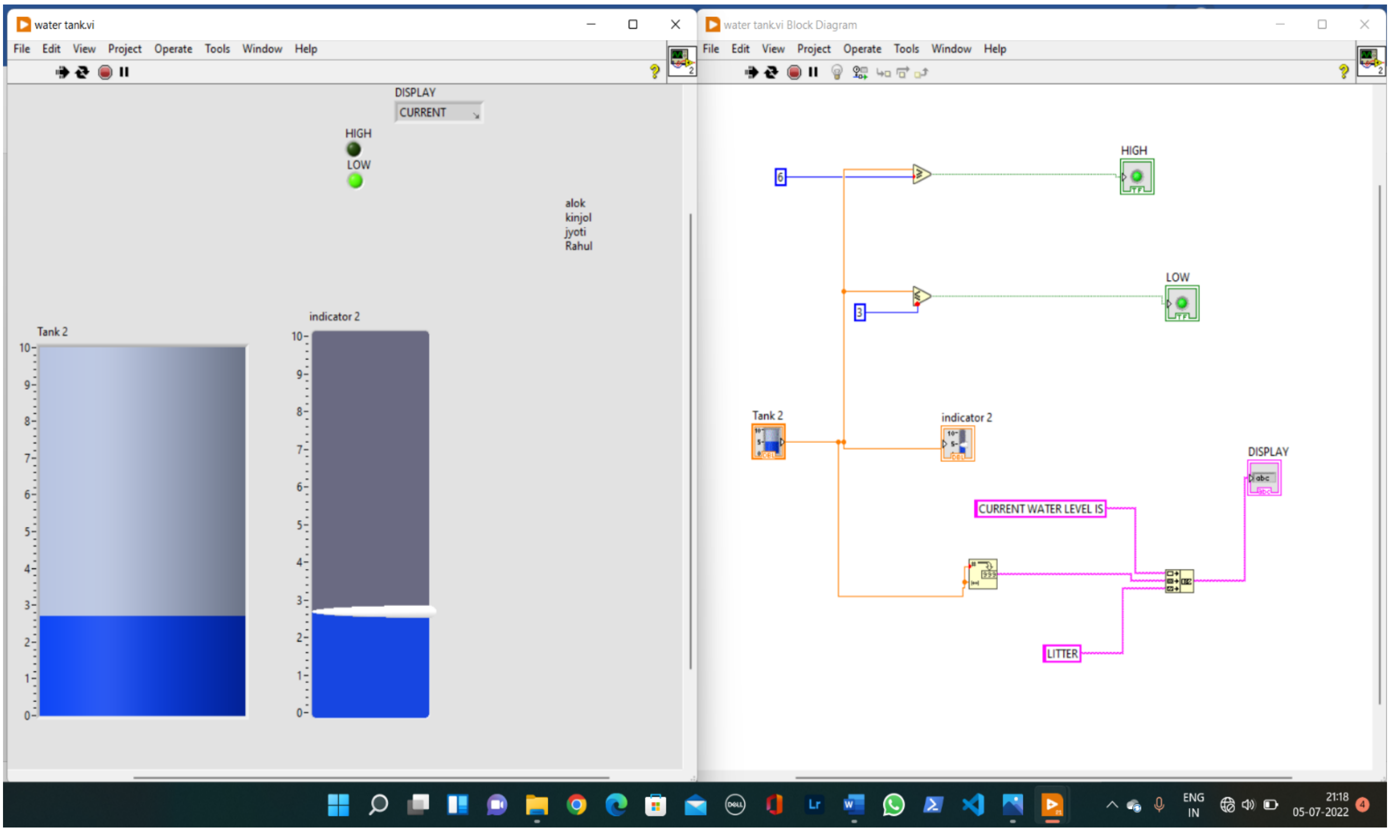
7. Now, go to the Controls-> Modern->comparison->greater or equal and connect the output to the high and create a constant. After that go to the Controls-> Modern->comparison->less or equal and connect the output to the low and create a constant.

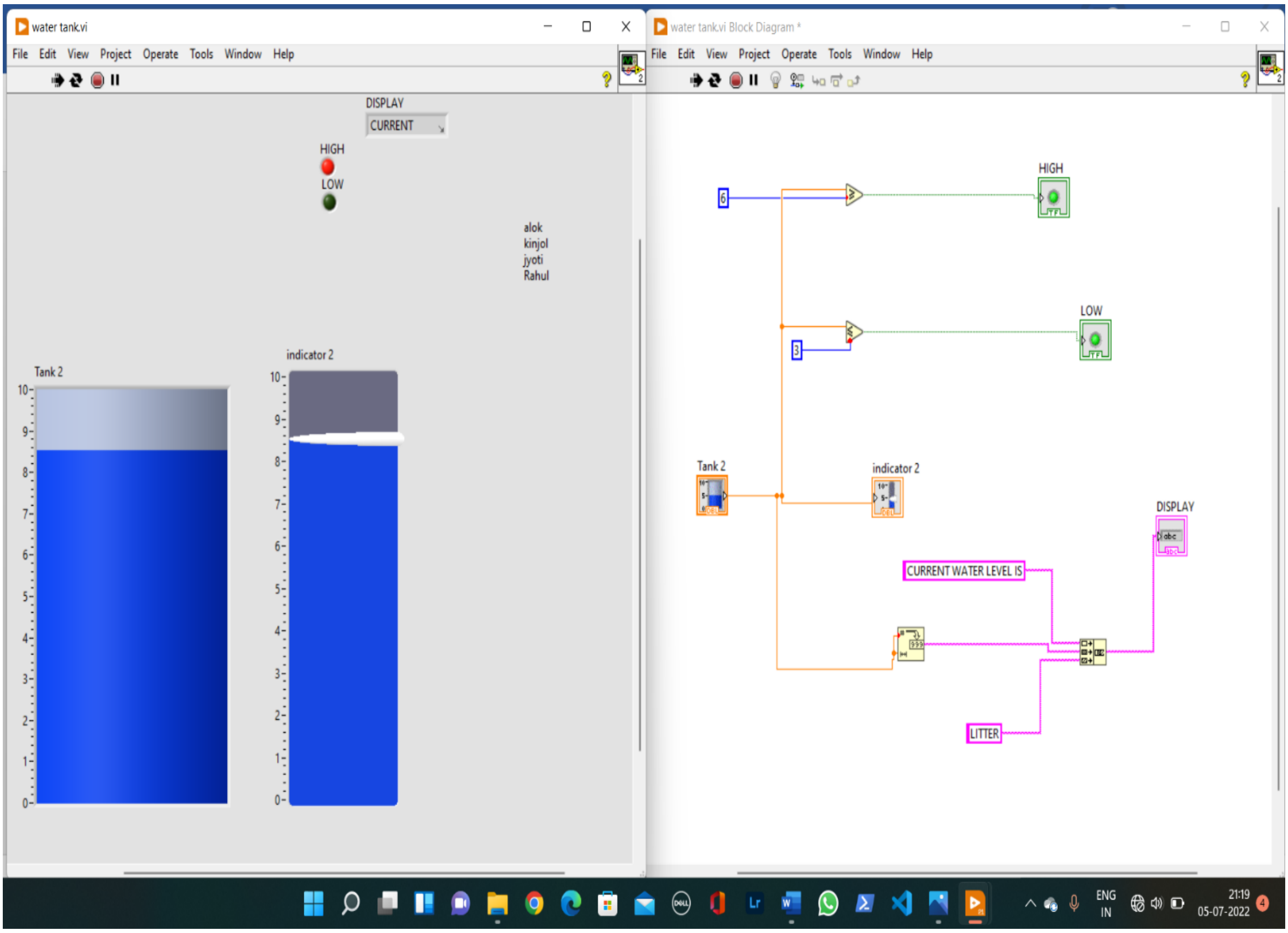
8. Now we use a function which will convert number to string for that go to function->string->number to string->number to decimal string.

9. For display the information current water level we take a concatenate string.

10. Now run the program.

RESULT





CONCLUSION

When we run the program if the level of water is above 6 L the green LED will glow and if the level of water is below 3 L the red LED will glow and the current water level will appear on the display. Vertical pointer is used to control the level of the water in the tank.

EXPERIMENT NO. 3

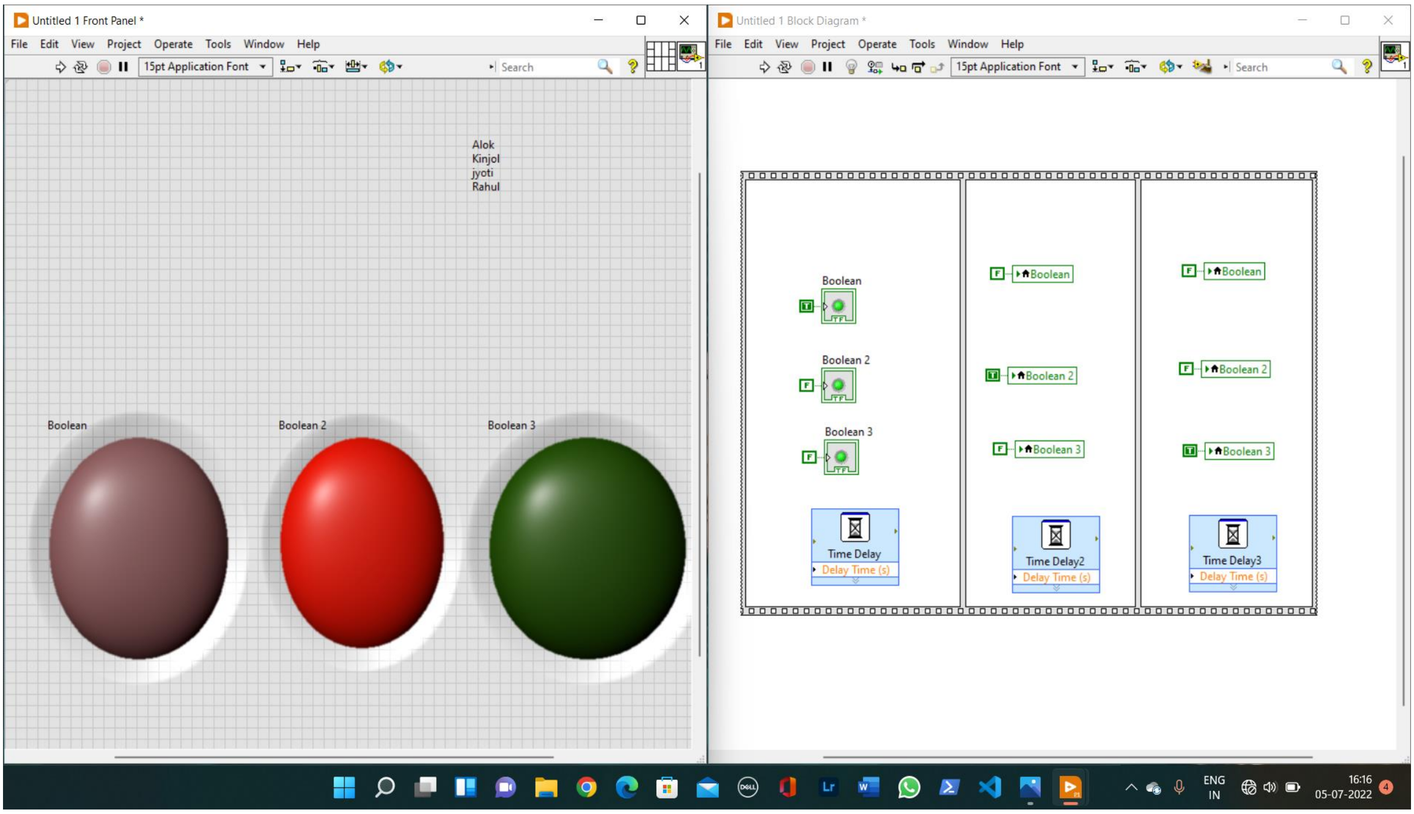
AIM OF THE EXPERIMENT: Design of a traffic light signal

OBJECTIVE: TO develop a traffic signal using LabView

SOFTWARE USED: LABVIEW 2019

LABVIEW COMPONENT: LEDS, FLAT SQUENCE

BLOKDIAGRAM:



PROCEDURE:

1. From the Boolean section, choose three leds and place them in the control panel in vertical order.

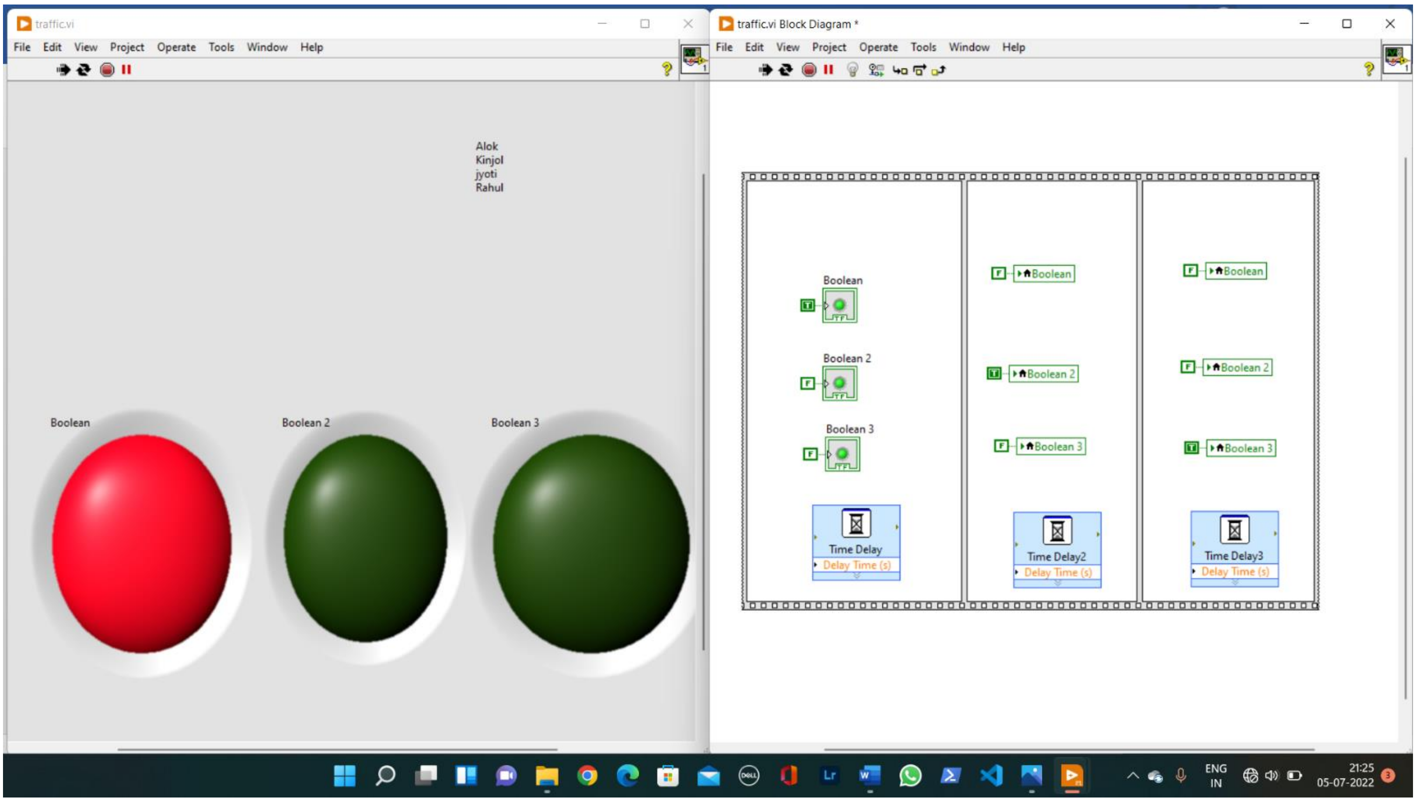
2. Encapsulate the leds within the 'flat sequence' in the block diagram panel. Add two more frames after that.

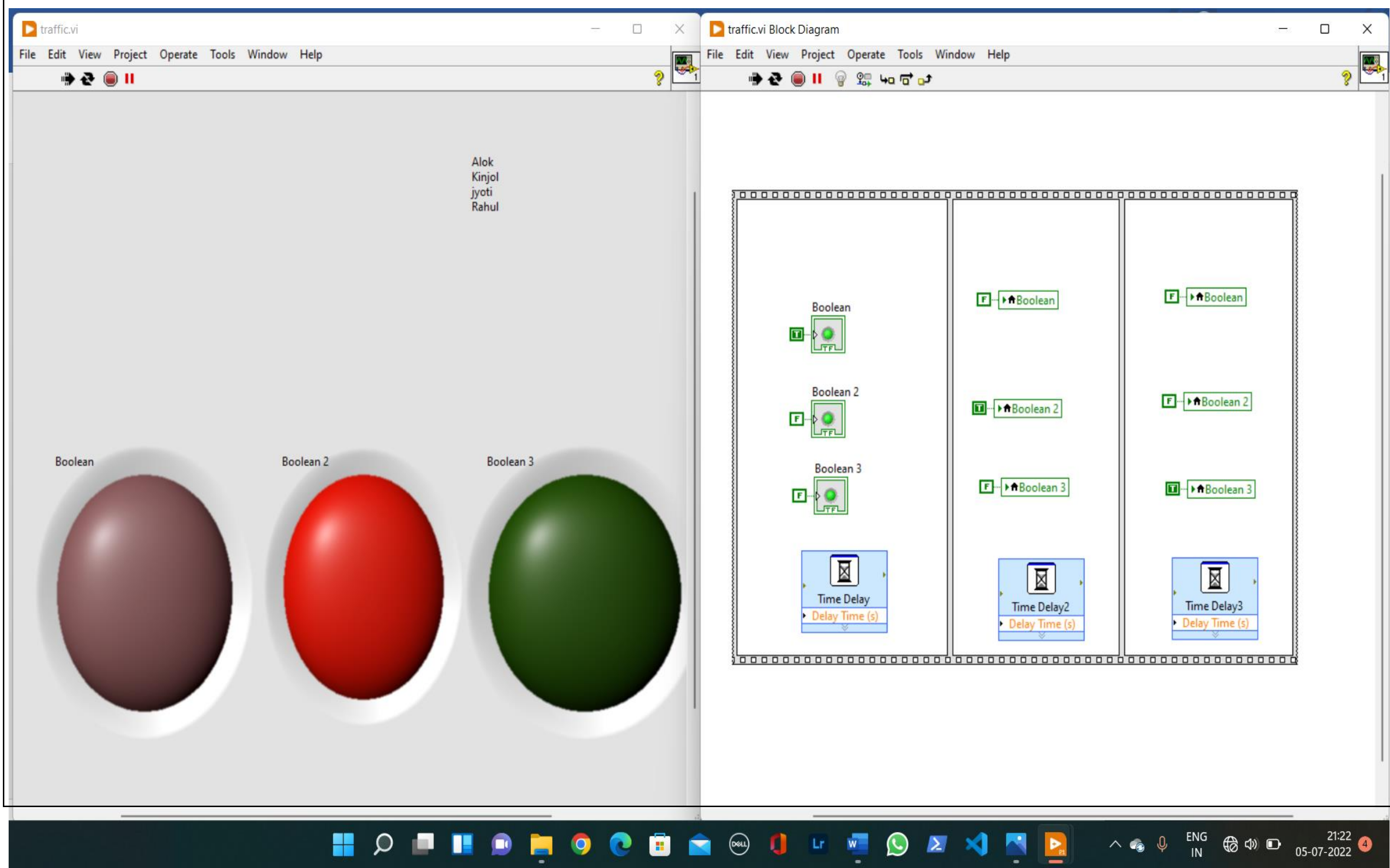
3.In the frames, place the local variables of the leds and change the constants to 'T' and 'F'.

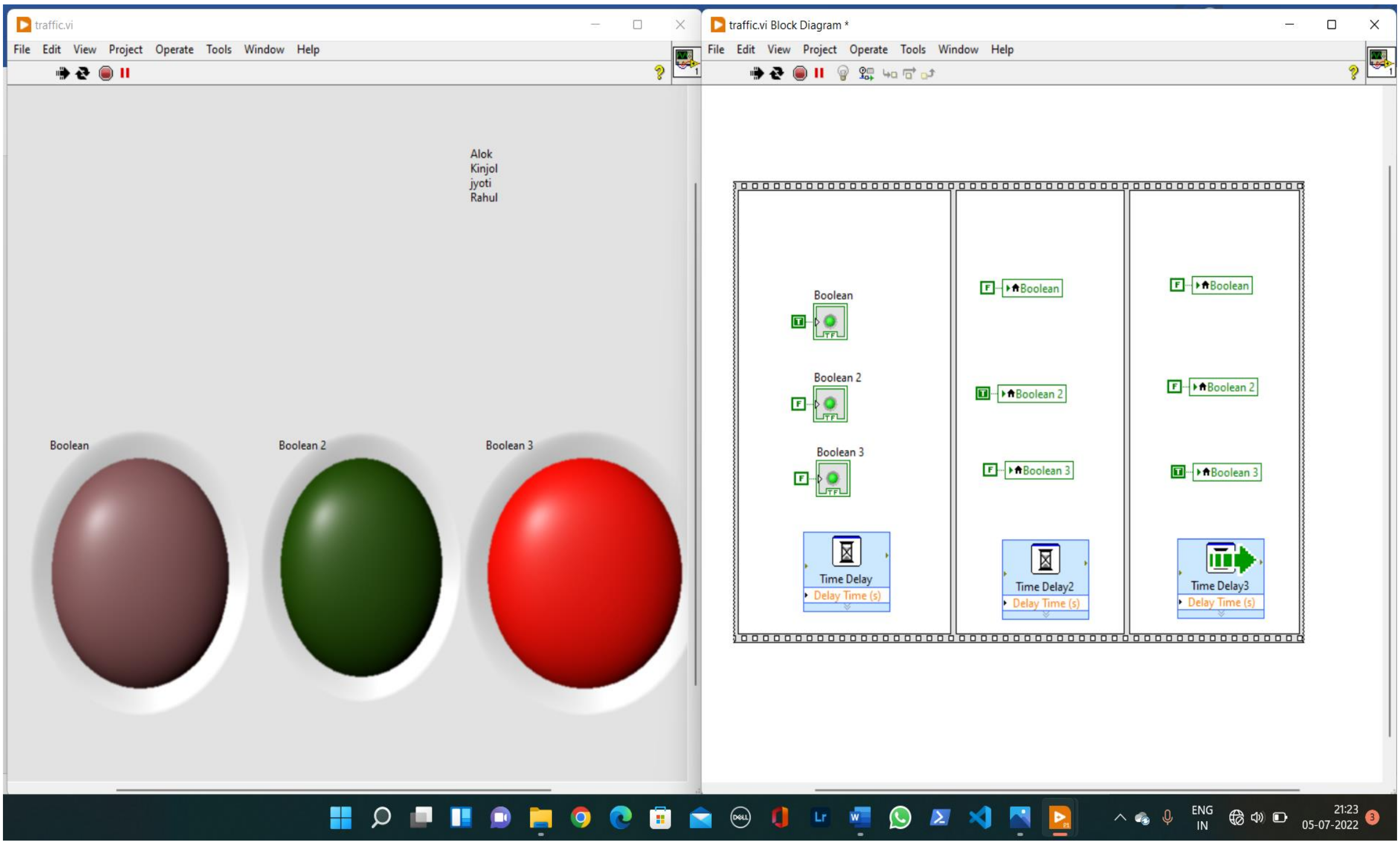
4.Then add the time delay block to each frame and set delay as 2 sec.

5.Now, run the program.

RESULT


traffic.vi
File Edit View Project Operate Tools Window Help
Alok
Kinjol
jyoti
Rahul
Boolean
Boolean 2
Boolean 3
traffic.vi Block Diagram *
Time Delay
Delay Time (s)
Time Delay2
Time Delay3
ENG
IN
21:25
05-07-2022


traffic.vi
File Edit View Project Operate Tools Window Help
Alok
Kinjol
jyoti
Rahul
Boolean
Boolean 2
Boolean 3
traffic.vi Block Diagram
Time Delay
Delay Time (s)
Time Delay2
Time Delay3
ENG
IN
21:22
05-07-2022



CONCLUSION:

When we run the program, at first the red light glows then after a delay of 2sec the yellow glows and finally after another 2 sec delay, thegreen glows.

EXPERIMENT NO: 04

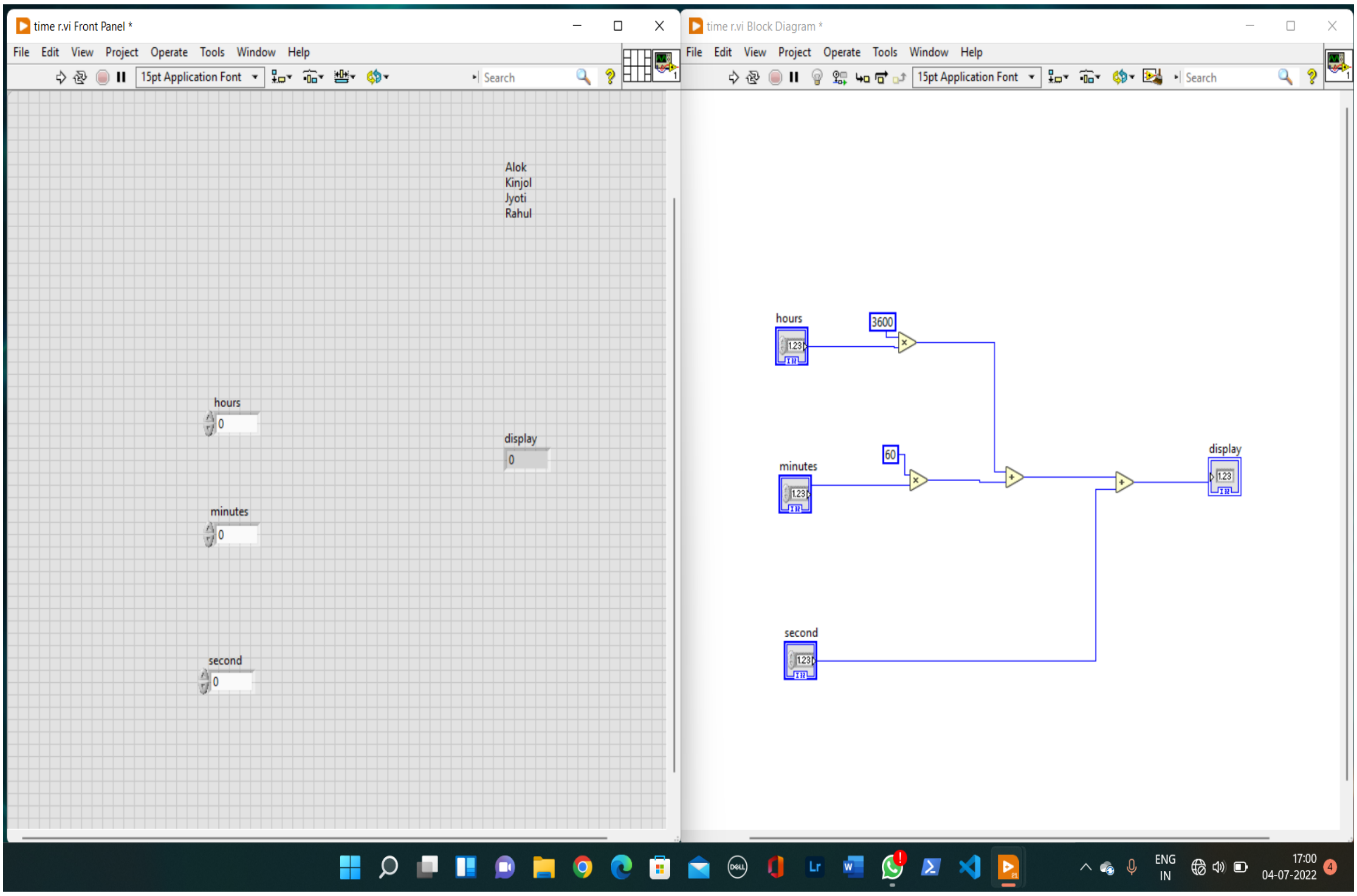
AIM OF THE EXPERIMENT: Designing of timer circuit

OBJECTIVE: To develop a simple timer in the labview and display the time count in LabVIEW

SOFTWARE USED: LABVIEW 2019

LABVIEW COMPONENTS: Adder, Multiplier ,Numeric control

BLOCK DIAGRAM



PROCEDURE

1. Go to the front panel and right click on it.

2. Now go to numeric>numeric control.

3. Select this block and placed it on the front panel.

4. Changed it name as hours, minutes, and second respectively.

5. Now right click on the panel and go to numeric and select adder.

6. Place the adder as shown in the block diagram.

7. Now again select multiplexer by going to numeric.

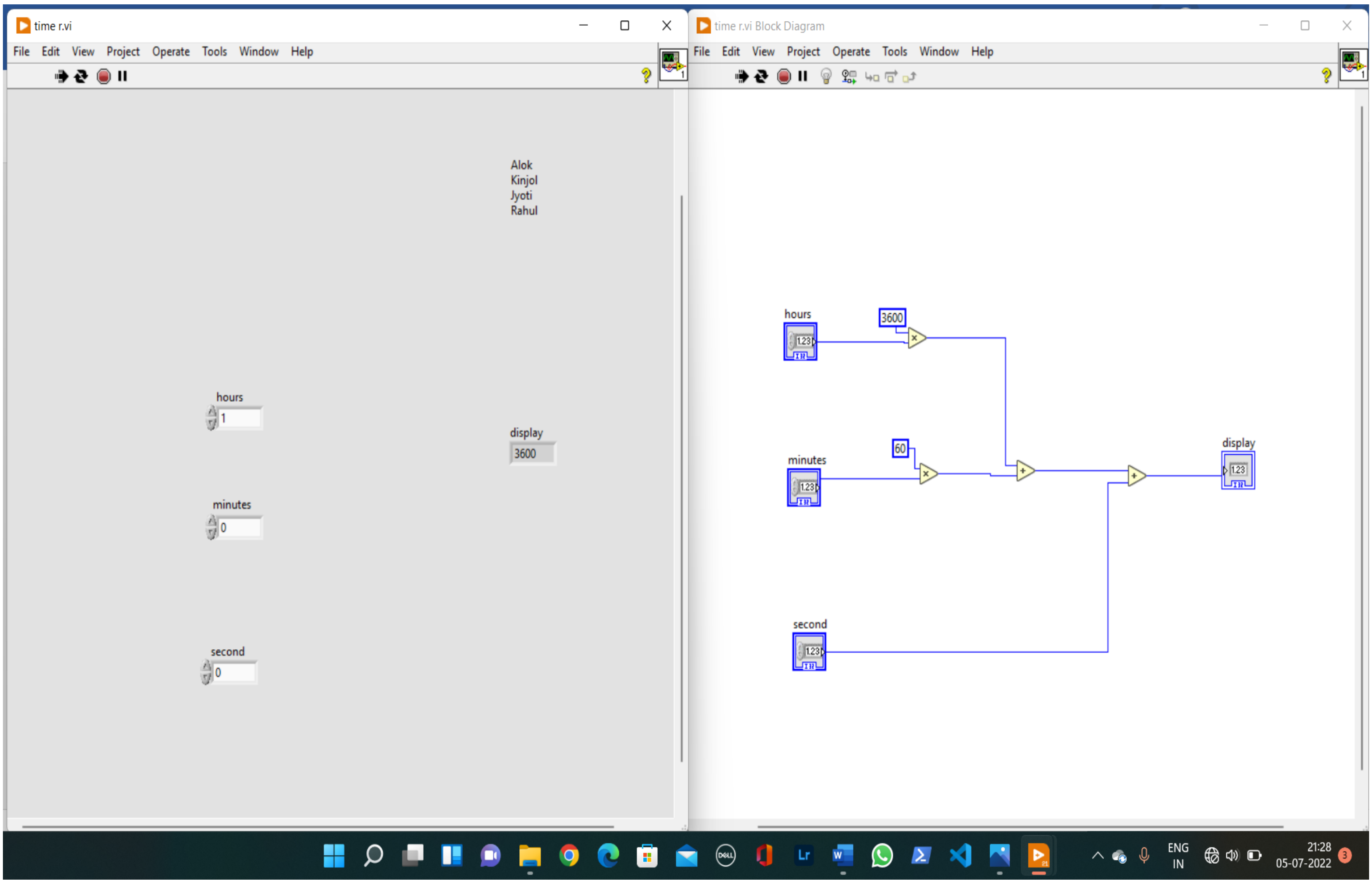
8. Place the multiplexer as shown in the block diagram.

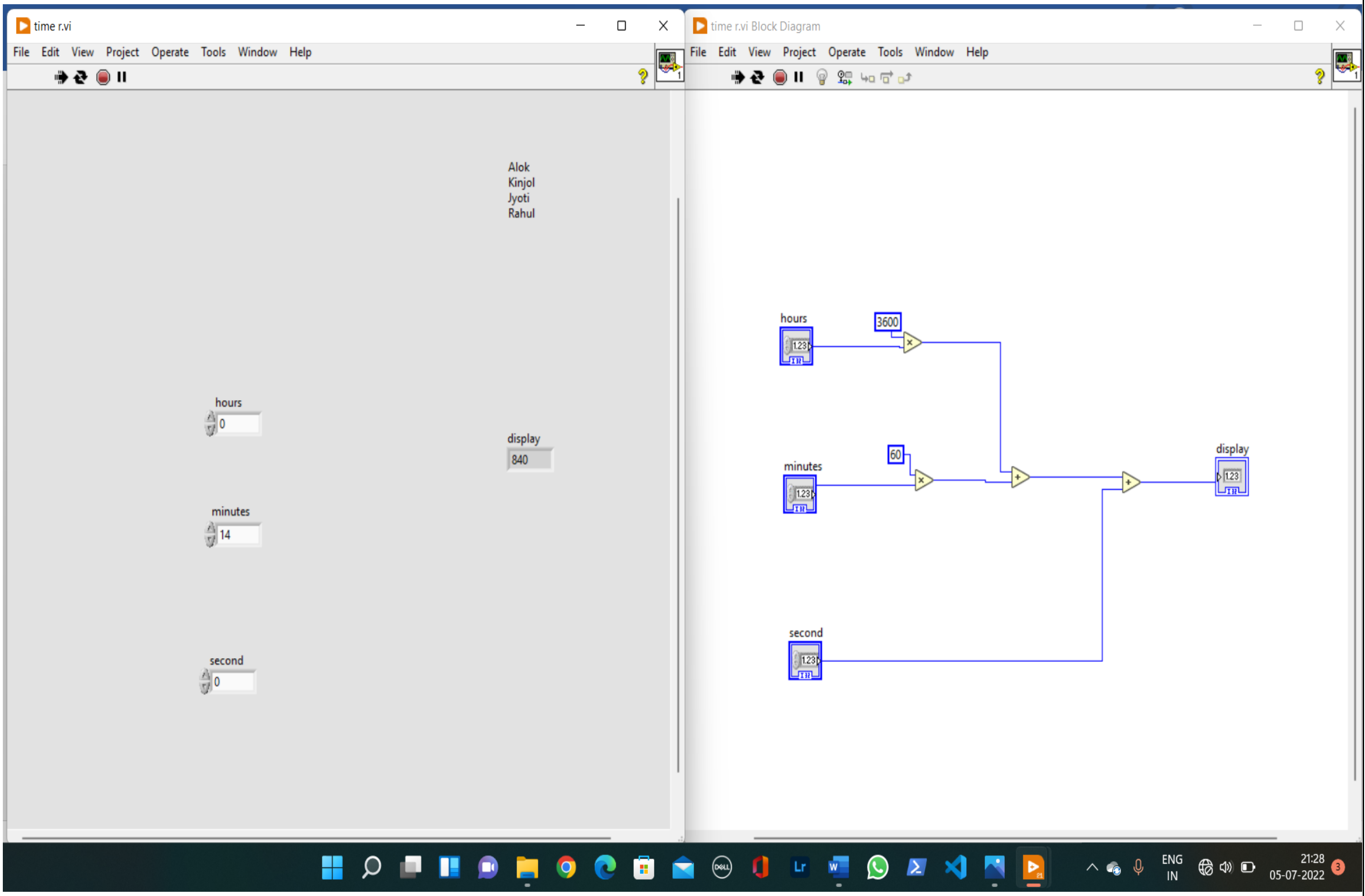
9. Now connect adder, multiplexer and numeric control as shown in the block diagram.

10. Now right click on the multiplexer and select 'constant' and place the value 3600 and 60 respectively.

11. Now run the program.

RESULT





CONCLUSION

When we give the time second it converts into hours, minutes and seconds.

EXPERIMENT NO: 05

AIM OF THE EXPERIMENT: Interfacing of 8085 microprocessor

OBJECTIVE: Interfacing/control of stepper motor motion in clockwise & counter clockwise direction by using 8085 microprocessors

Components Used: 8085 microprocessor kit, stepper motor, interfacing units, regulated power supply, wire cable

Stepper Motor

A stepper motor is a device that translates electrical pulses into mechanical movement in steps of fixed step angle.

- The stepper motor rotates in steps in response to the applied signals.
- It is mainly used for position control.
- It is used in disk drives, dot matrix printers, plotters and robotics and process control circuits.

1. **Structure**

   Stepper motors have a permanent magnet called rotor (also called the shaft) surrounded by a stator. The most common stepper motors have four stator windings that are paired with a centre-tap. This type of stepper motor is commonly referred to as a four-phase or unipolar stepper motor. The centre tap allows a change of current direction in each of two coils when a winding is grounded, thereby resulting in a polarity change of the stator.

2. **Interfacing**

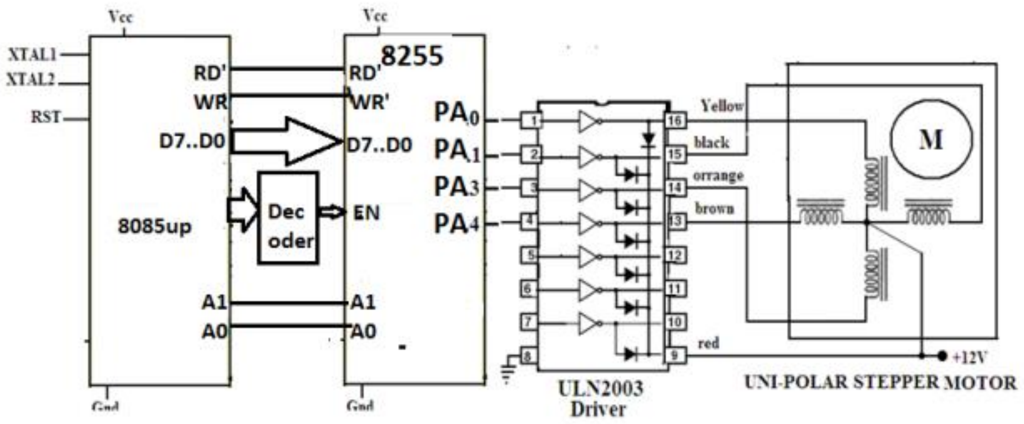
   Even a small stepper motor requires a current of 400 mA for its operation. But the ports of the microcontroller cannot source this much amount of current. If such a motor is directly connected to the microprocessor/microcontroller ports, the motor may draw large current from the ports and damage it. So, a suitable driver circuit is used with the microprocessor/microcontroller to operate the motor.

3. **Motor Driver Circuit (ULN2003)**
   Stepper motor driver circuits are available readily in the form of ICs. ULN2003 is one such driver IC which is a High-Voltage High-Current Darlington transistor array and can give a current of 500mA.This current is sufficient to drive a small stepper motor. Internally, it has protection diodes used to protect the motor from damage due to back emf and large eddy currents. So, this ULN2003 is used as a driver to interface the stepper motor to the microcontroller

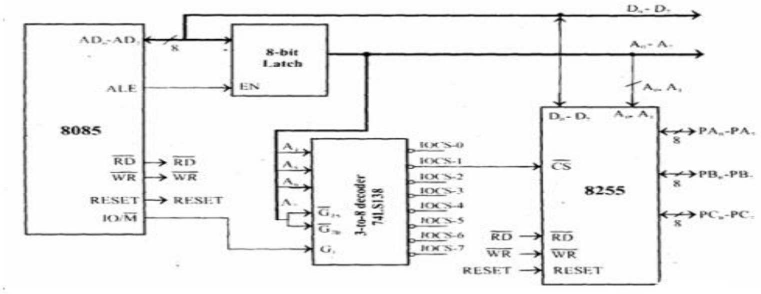
Operation

The important parameter of a stepper motor is the step angle. It is the minimum angle through which the motor rotates in response to each excitation pulse. In a four phase motor if there are 200 steps in one complete rotation then then the step angle is 360/200 = 1.8O . So to rotate the stepper motor we have to apply the excitation pulse. For this the controller should send a hexa decimal code through one of its ports. The hex code mainly depends on the construction of the stepper motor. So, all the stepper motors do not have the same Hex code for their rotation. (refer the operation manual supplied by the manufacturer.) For example, let us consider the hex code for a stepper motor to rotate in clockwise direction is 77H , BBH , DDH and EEH. This hex code will be applied to the input terminals of the driver through the assembly language program. To rotate the stepper motor in anti-clockwise direction the same code is applied in the reverse order

Stepper Motor interface - Schematic Diagram for (8085)



Detailed Connection diagram between 8085 and 8255



ASSEMBLY LANGUAGE PROGRAM (8085)

```
Main : MVI A, 80          ; 80H → Control word to configure PA,PB,PC in O/P
OUT CWR_Address           ; Write control word in CWR of 8255
MVI A, 77                 ; Code for the Phase 1
OUT PortA_Address         ; sent to motor via port A of 8255 ;
CALL DELAY                ; Delay subroutine
MVI A, BB                 ; Code for the Phase II
OUT PortA_Address         ; sent to motor via port A of 8255
CALL DELAY                ; Delay subroutine.
MVI A, DD                 ; Code for the Phase III
OUT PortA_Address         ; sent to motor via port A of 8255;
```

```
CALL DELAY                 ; Delay subroutine
MVI A, EE H                ; Code for the Phase 1
OUT PortA_Address          ; sent to motor via port A of 8255
CALL DELAY                 ; Delay subroutine
JMP MAIN                   ; Keep the motor rotating continuously
```

DELAY Subroutine

```
MVI C, FF                  ; Load C with FF -- Change it for the speed variation
LOOP1:                     MVI D,FF ; Load D with FF
LOOP2: DCR D
JNZ LOOP2
DCR C
JNZ LOOP1RET                        ; Return to main program .
```