3.Create a program that blinks the LED on the development board using MBED software

Aim: To write a program for blink led using tinker Cad.

Components Used:

1.Arduino Uno

2. LED(Light Emitting Diode)

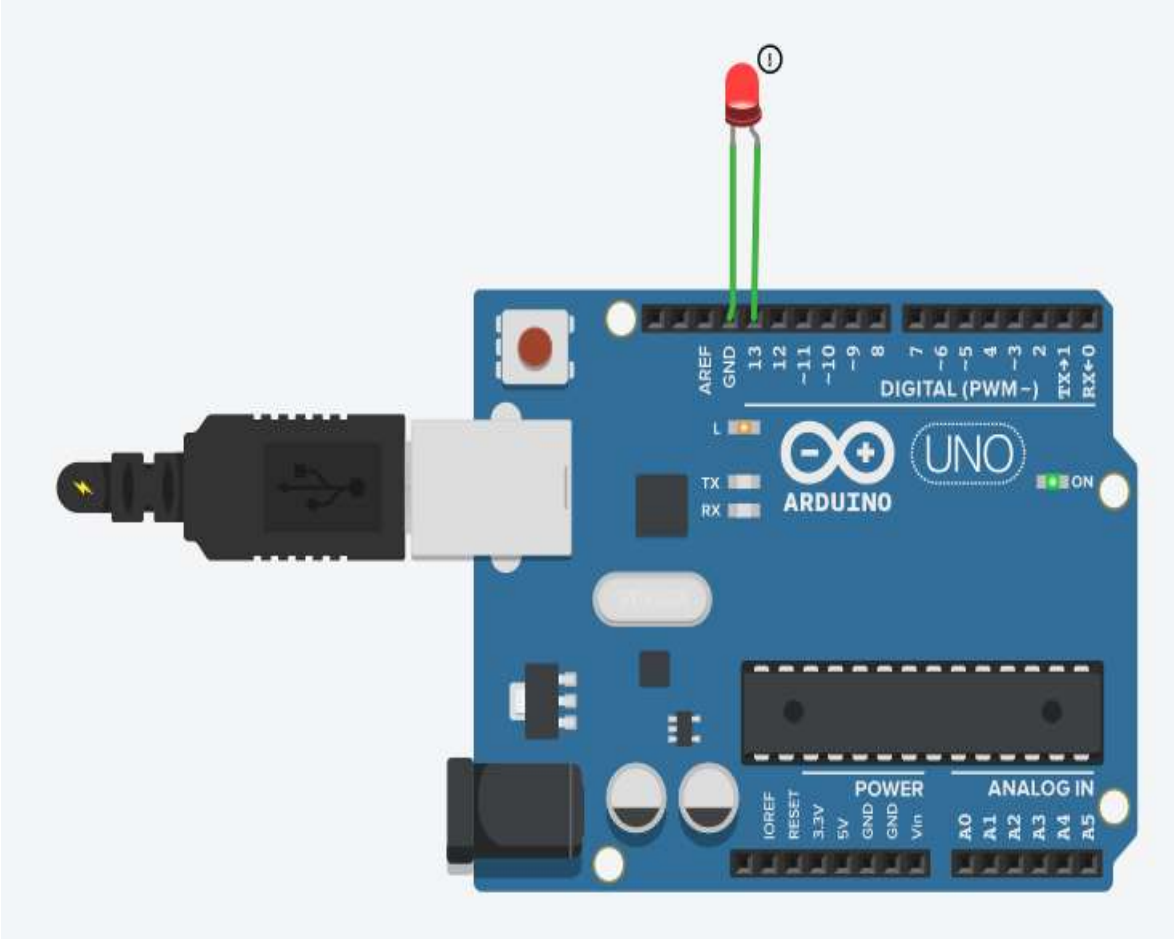
Procedure:

1.Get the Arduino uno board from the components

2.Get the LED from the components

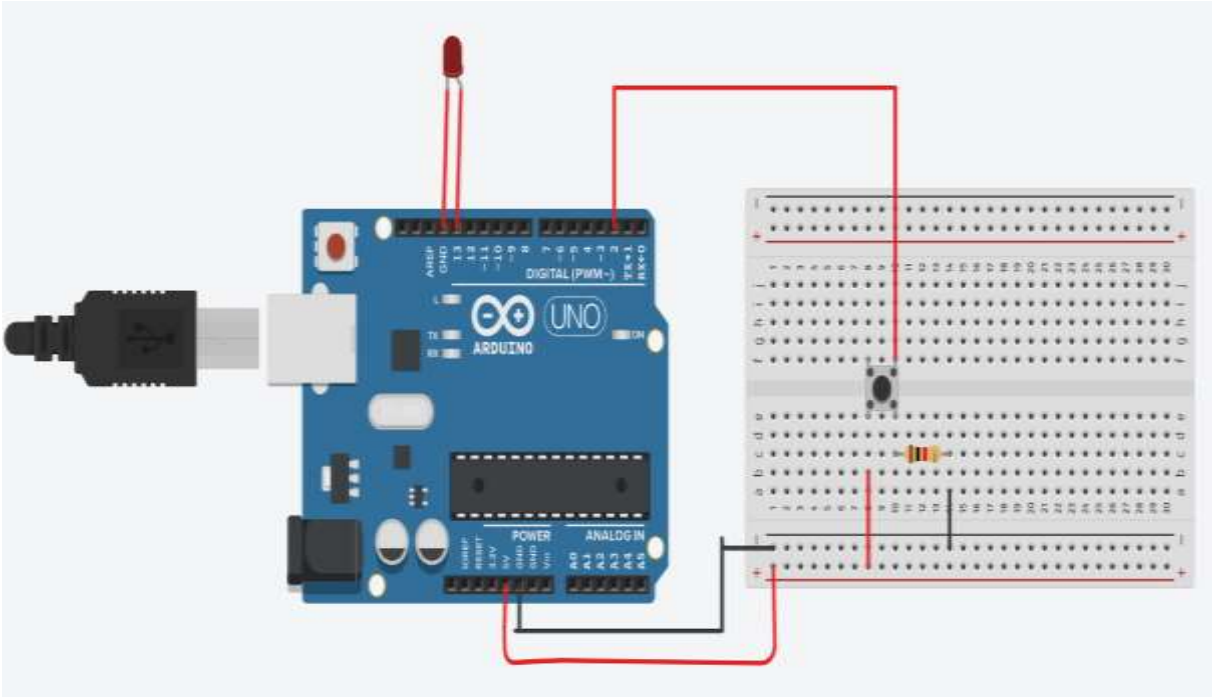
3.LED has two side which is positive (anode) and Negative(Cathode).Negative side is connected to the Ground(GND).Positive side is connected to Digital pin 13 of Arduino .

3.1 Code:



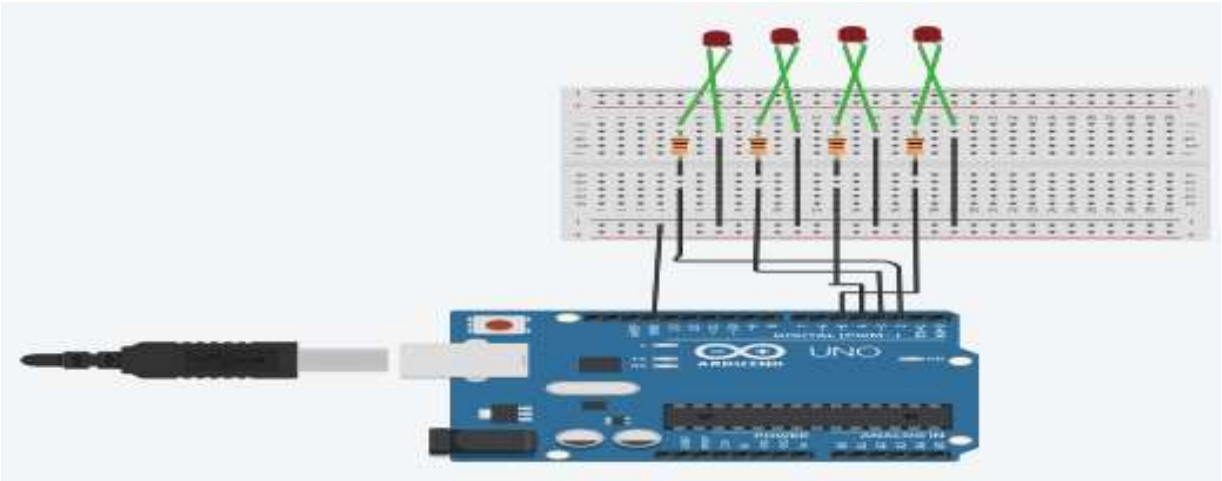

```
void setup()
{
pinMode(13, OUTPUT);

}
void loop()
{
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

3.2.Through Button blink LED




```
void setup()
{
  pinMode(2, INPUT);
  pinMode(13,OUTPUT);
}
void loop()
{
  if(digitalRead(2)==1)
  {
    digitalWrite(13,HIGH);
  }
  else
  {
    digitalWrite(13,LOW);
  }
}
```

3.3. Led with digital counter



CODE:

```
int pin2=2;
int pin3=3;
int pin4=4;
int pin5=5;
int stime=500;
void setup()
{
  pinMode(pin2,OUTPUT);
  pinMode(pin3,OUTPUT);
  pinMode(pin4,OUTPUT);
  pinMode(pin5,OUTPUT);
}
void loop()
{
  digitalWrite(pin2,LOW);
  digitalWrite(pin3,LOW);
  digitalWrite(pin4,LOW);
  digitalWrite(pin5,LOW);
  delay(stime);
  digitalWrite(pin2,LOW);
  digitalWrite(pin3,LOW);
  digitalWrite(pin4,LOW);
  digitalWrite(pin5,HIGH);
  delay(stime);
  digitalWrite(pin2,LOW);
  digitalWrite(pin3,LOW);
  digitalWrite(pin4,HIGH);
  digitalWrite(pin5,LOW);
  delay(stime);
  digitalWrite(pin2,LOW);
  digitalWrite(pin3,LOW);
  digitalWrite(pin4,HIGH);
  digitalWrite(pin5,HIGH);
  delay(stime);
  digitalWrite(pin2,LOW);
  digitalWrite(pin3,HIGH);
  digitalWrite(pin4,LOW);
  digitalWrite(pin5,LOW);
  delay(stime);
  digitalWrite(pin2,LOW);
  digitalWrite(pin3,HIGH);
  digitalWrite(pin4,LOW);
  digitalWrite(pin5,HIGH);
  delay(stime);
  digitalWrite(pin2,LOW);
  digitalWrite(pin3,HIGH);
  digitalWrite(pin4,HIGH);
```

```
  digitalWrite(pin5,LOW);
  delay(stime);
  digitalWrite(pin2,LOW);
  digitalWrite(pin3,HIGH);
  digitalWrite(pin4,HIGH);
  digitalWrite(pin5,HIGH);
  delay(stime);
  digitalWrite(pin2,HIGH);
  digitalWrite(pin3,LOW);
  digitalWrite(pin4,LOW);
  digitalWrite(pin5,LOW);
  delay(stime);
  digitalWrite(pin2,HIGH);
  digitalWrite(pin3,LOW);
  digitalWrite(pin4,LOW);
  digitalWrite(pin5,HIGH);
  delay(stime);
  digitalWrite(pin2,HIGH);
  digitalWrite(pin3,LOW);
  digitalWrite(pin4,HIGH);
  digitalWrite(pin5,LOW);
  delay(stime);
  digitalWrite(pin2,HIGH);
  digitalWrite(pin3,LOW);
  digitalWrite(pin4,HIGH);
  digitalWrite(pin5,HIGH);
  delay(stime);
  digitalWrite(pin2,HIGH);
  digitalWrite(pin3,HIGH);
  digitalWrite(pin4,LOW);
  digitalWrite(pin5,LOW);
  delay(stime);
  digitalWrite(pin2,HIGH);
  digitalWrite(pin3,HIGH);
  digitalWrite(pin4,LOW);
  digitalWrite(pin5,HIGH);
  delay(stime);
  digitalWrite(pin2,HIGH);
  digitalWrite(pin3,HIGH);
  digitalWrite(pin4,HIGH);
  digitalWrite(pin5,LOW);
  delay(stime);
  digitalWrite(pin2,HIGH);
  digitalWrite(pin3,HIGH);
  digitalWrite(pin4,HIGH);
  digitalWrite(pin5,HIGH);
  delay(stime);
}
```

4. Pick one-one from the available sensors and actuators and find or create code that will display the sensed data on the pc

A) Analog potentiometer

Aim : To write a program for analog potentiometer using tinker Cad.

Procedure:

1.Get the Arduino Uno board from the components

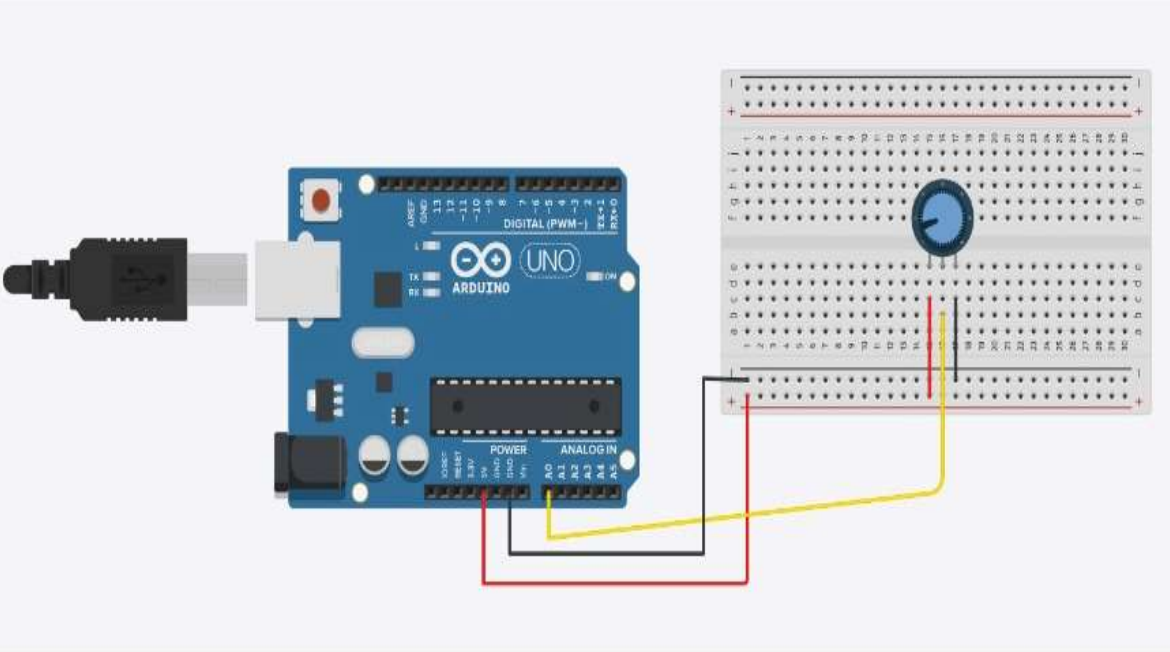
2.Get the bread board from the components

3.Get the potentiometer from component .The potentiometer has 3 pins. First is connected to 5 v .Second pin is connected to A0 in analog pin.Third pin is connected into a gnd

Code :

```
Int pot=A0;
Void setup()
{
Serial.begin(9600);
}
Void loop()
{
  Int potvalue=analogRead(pot);
  Serial.print("pot value");
  Serial.println(potvalue);
  Delay(1);
}
```

Prototype :



Output :

Pot value818
Pot value777
Pot value716
Pot value696
Pot value675
Pot value675

Result : The above experiment is executed successfully

B) Reading sensor

Aim : To write a program for reading sensors using tinker Cad.

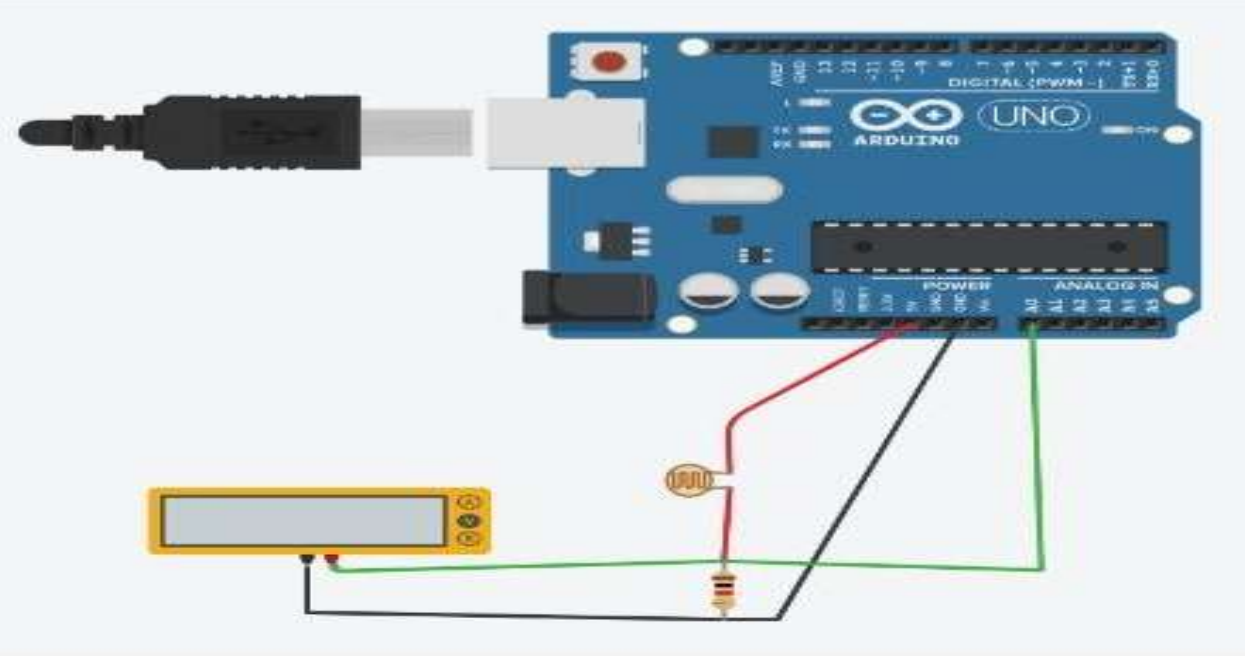
Procedure:

1.Get the Arduino Uno board from the components
2.Get the photoresistor from the components
3.Get the resistor from the components
4. Get the multimeter from the components

Code :

```
Void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
}
Void loop()
{
   Int lightvalue=analogRead(A0);
   Serial.println(lightvalue);
    Delay(1000);
}
```

Prototype:



Output :

6
379
526
640
658
663
654
476

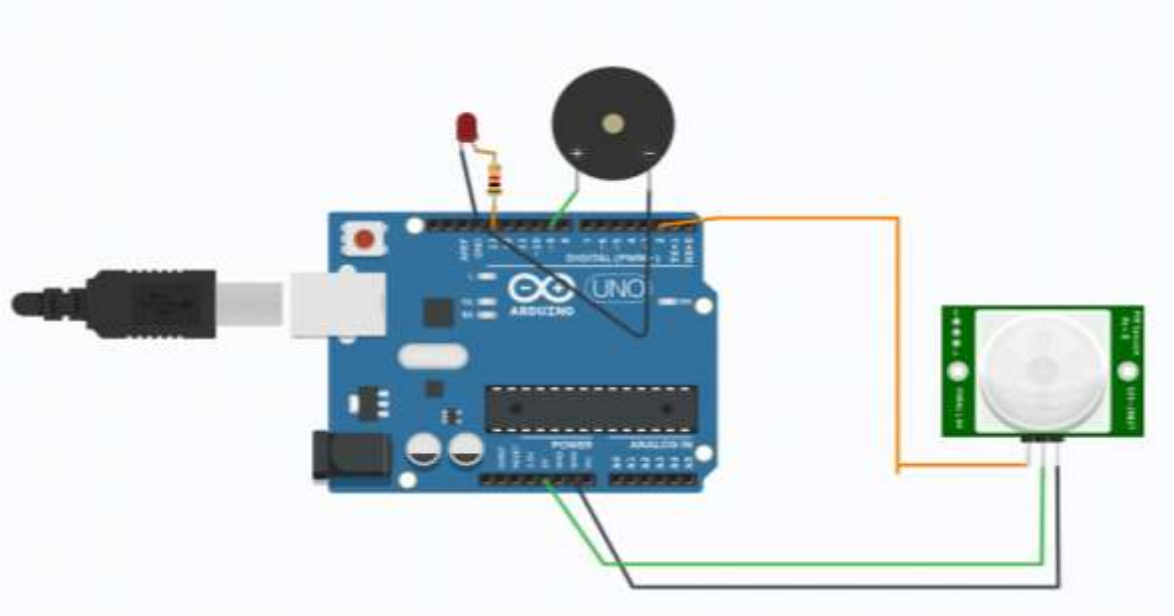
Result : The above experiment is executed successfully

5.Create a program that displays data from the sensor in regular intervals in a compact format.

Aim: To create a program that displays data from the sensor in regular intervals in a compact format.

Algorithm:

1. Set up pin modes for input (pin 2) and outputs (pins 13 and 9).
2. Enter an infinite loop.
3. Read the state of pin 2.
4. If the state is high, turn on the LED (pin 13) and play a tone (523 Hz) on pin 9.
5. If the state is not high, turn off the LED and stop the tone.
6. Delay for 1 millisecond.
7. Repeat from step 3.



```
void setup()
{
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
  pinMode(9, OUTPUT);
}
void loop()
{
  if (digitalRead(2) >= HIGH) {
    digitalWrite(13, HIGH);
    tone(9, 523, 1000); // play tone 60 (C5 = 523 Hz)
  } else {
    digitalWrite(13, LOW);
    noTone(9);
  }
  delay(1); // Wait for 1 millisecond(s)
}
```

Result: To create a program that displays data from the sensor in regular intervals in a compact format is executed.

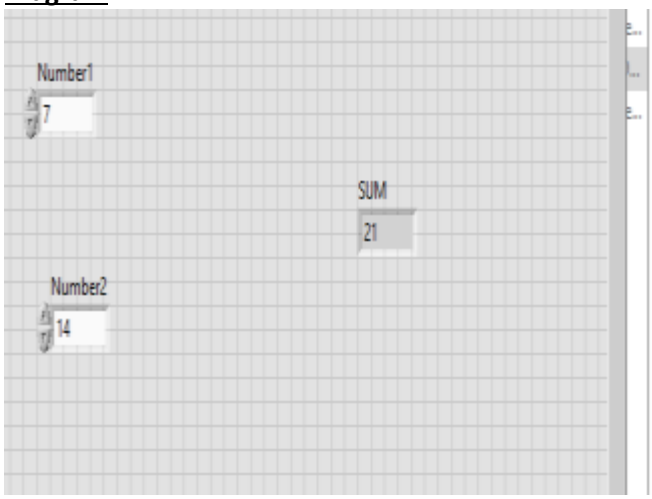
6. To perform simple addition using embedded components in Lab view software.

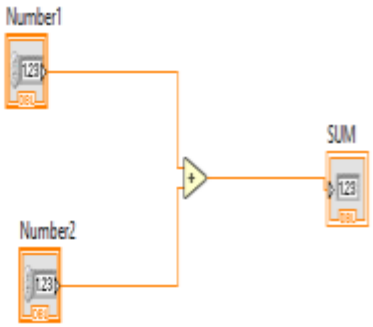
AIM: To build a VI that performs addition of two numbers and displays the result using LABVIEW.

ALGORITHM

1.Open labview software
2.Open blank VI.
3.Right click on the front panel window to open the control palette.
4.Insert the numeric controls as a input from the control palette.
5.Right click on the block diagram to open the function palette.
6.Add the Adder tool from the numeric sub palette from the function palette.
7.Insert the numeric indicator as an output from the control palette.
8.Connect all the terminals in the block diagram window.
9.Enter inputs in the front panel and click run to display the result.

Diagram





RESULT: Thus, a VI performs addition operation and displays the result using LABVIEW.

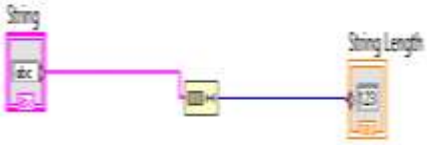
7. To perform string operations using embedded components in Lab view software.

a) STRING LENGTH

Aim: To build a VI that performs string length operation using embedded components and displays the result using LABVIEW.

Algorithm:

1.Create a new VI.
2.Drag and drop a string control and a numeric indicator onto the front panel.
3.Connect the string control to the input of the numeric indicator.
4.Right-click on the numeric indicator and select "Properties."
5.Set the display format of the numeric indicator to "Decimal."
6.Add a String Length property node to the block diagram.
7.Connect the string control to the input of the String Length property node.
8.Connect the output of the String Length property node to the numeric indicator.
9.Run the VI and enter the input string in the string control.
10.Observe the length of the string on the numeric indicator.



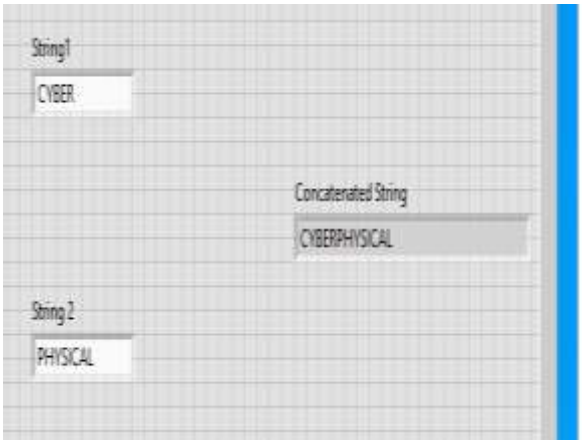
Result: Thus, a VI performs string length operation and displays the result using LABVIEW

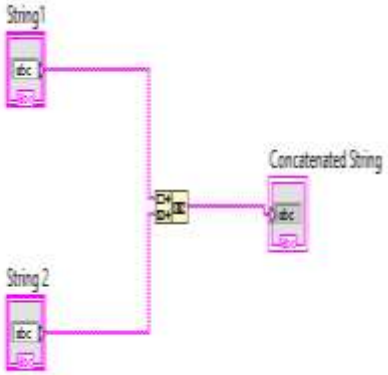
b) STRING CONCATENATION

Aim: To build a VI that performs string concatenation operation using embedded components and displays the result using LABVIEW.

Algorithm:

1.Create a new VI.
2.Drag and drop two string controls and a string indicator onto the front panel.
3.Connect the two string controls to the inputs of the string indicator.
4.Add a Concatenate Strings function or use the concatenation operator.
5.Connect the two string controls to the inputs of the Concatenate Strings function or use the concatenation operator.
6.Connect the output of the Concatenate Strings function or the concatenation operator to the string indicator.
7.Run the VI and enter the input strings in the string controls.
8.Observe the concatenated string on the string indicator.





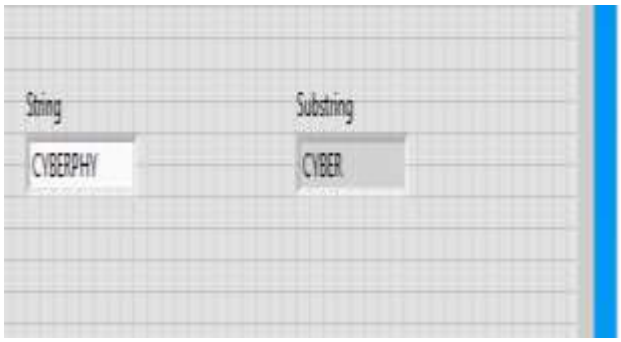
Result: Thus, a VI performs string concatenation operation and displays the result using LABVIEW

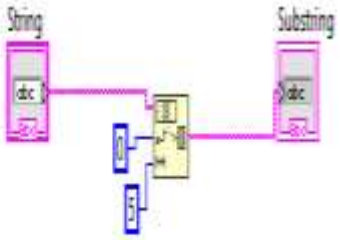
c) SUBSTRING

Aim: To build a VI that performs substring operation using embedded components and displays the result using LABVIEW.

Algorithm:

1.Create a new VI.
2.Drag and drop a string control, two numeric controls, and a string indicator onto the front panel.
3.Connect the string control, starting index control, and length control to the inputs of the string indicator.
4.Add a Substring function to the block diagram.
5.Connect the string control to the input of the Substring function.
6.Connect the starting index control to the "Start" input of the Substring function.
7.Connect the length control to the "Length" input of the Substring function.
8.Connect the output of the Substring function to the string indicator.
9.Run the VI and enter the input string, starting index, and length.
10.Observe the extracted substring on the string indicator.





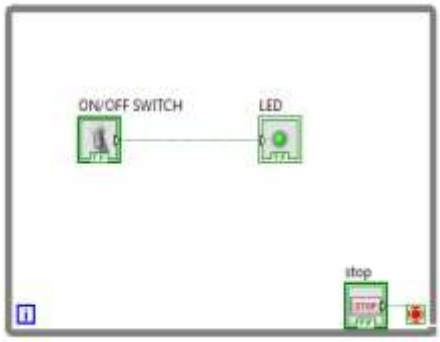
Result: Thus, a VI performs substring operation and displays the result using LABVIEW

8. To perform LED ON/OFF switch using Lab view software.

Aim: To build a VI that performs led on/off switch operation using embedded components and displays the result using LABVIEW.

Algorithm:

1.Connect the LED to the digital output pin of the microcontroller.
2.Open LabVIEW software and create a new VI (Virtual Instrument).
3.Add a digital output module to the block diagram.
4.configure it to control the digital output pin connected to the LED.
5.Add a Boolean switch to the front panel of the VI to control the LED ON/OFF state.
6.Connect the Boolean switch to the digital output module on the block diagram.
7.Run the VI and toggle the Boolean switch to turn the LED ON/OFF.
8.Observe the LED ON/OFF switch operation on the indicator.



Result: Thus, a VI performs led on/off switch operation and displays the result using LABVIEW

9. To Design Traffic Signal Light using embedded components using Lab view software.

Aim: To build a VI that performs Traffic signal light operation using embedded components and displays the result using LABVIEW.

Algorithm:-

1. Initialize the System:
   *set up the LabVIEW environment and ensure all necessary components (microcontroller, LED, etc.) are connected and functioning correctly.
2. Define Signal Timing:
   *Decide on the timing intervals for each phase of the traffic signal (eg: green, yellow red times for each direction).
3. Initialize Signal states:
   * Set the initial state of the traffic signal to a predefined starting configuration (eg: all red or a specific direction green).
4. Loop for Traffic signal control:
   *Enter a continuous loop that will handle the traffic signal control.
5. Check Input:

*Monitor the sensors or inputs to detect vehicle presence or traffic demands for different directions.

6. Implement Traffic signal logic.

*Based on the Input from the previous step, implement the traffic signal Logic to determine the appropriate signal states for each direction.

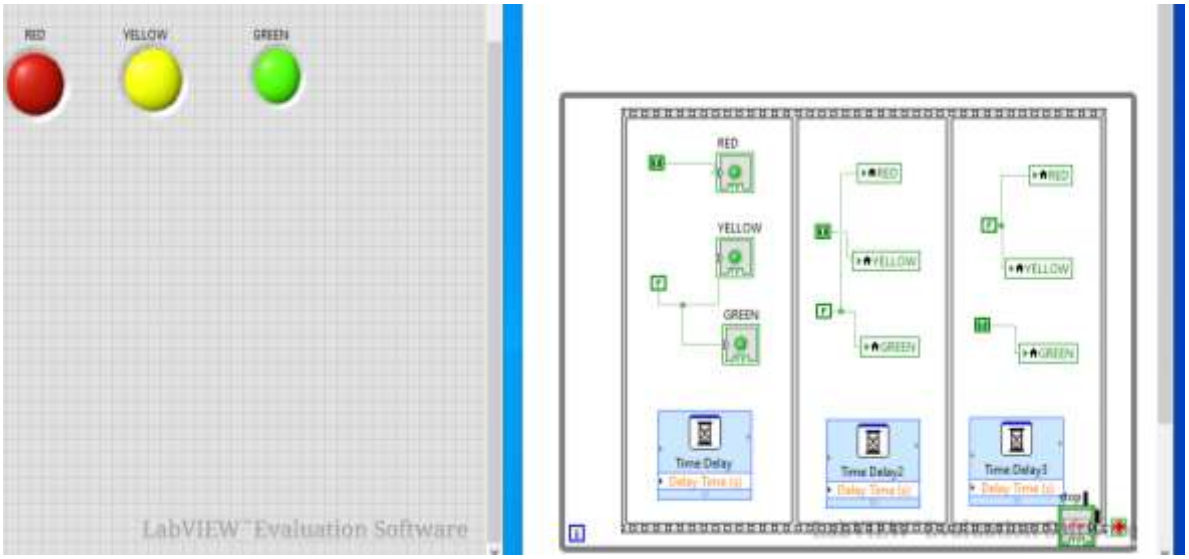
*Typical logic involves transitioning from green to yellow before changing to red and vice versa.

7. Update signal outputs:

8. wait for signed change

9. Repeat the loop.

10. End the program.



Result: To Design Traffic Signal Light using embedded components using Lab view software is implemented successfully.