Bahria University

Karachi Campus

A logo with text on it

Description automatically generated

LAB EXPERIMENT NO.

**5**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | Write a sketch to blink the 2 LEDs interfaced with Arduino at a different rate simultaneously. |
| 2 | Write a sketch to implement the one-way traffic light controller using FSM concepts. The sensor will work to sense the traffic on the road whose output will be the stimulus for the state transition. |

Submitted On:

29 December 2023

\_\_\_\_\_\_\_\_\_\_\_\_

(Date: DD/MM/YY)

**Task 1**

Write a sketch to blink the 2 LEDs interfaced with Arduino at a different rate simultaneously.

const int led1 = 3;

const int led2 = 5;

int led1state = HIGH;

int led2state = HIGH;

long previoustimeled1 = 0;

long previoustimeled2 = 0;

long led1interval = 1000;

long led2interval = 400;

void setup(){

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);}

void loop(){

  unsigned long currenttime = millis();

  if (currenttime-previoustimeled1 > led1interval){

    toggleled1();

    previoustimeled1=currenttime;}

  if (currenttime-previoustimeled2 > led2interval){

    toggleled2();

    previoustimeled2=currenttime;}}

void toggleled1(){

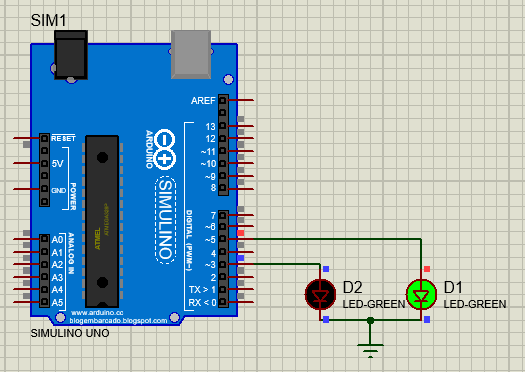
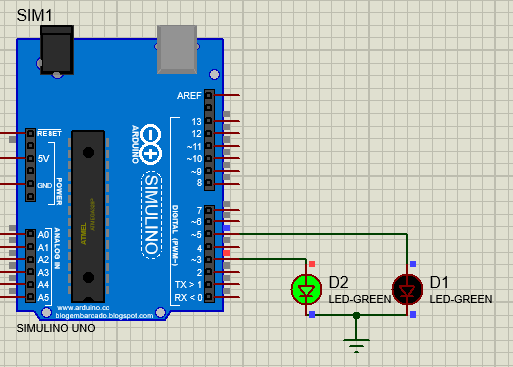
  led1state = (led1state == HIGH) ? LOW : HIGH;

  digitalWrite (led1, led1state);}

void toggleled2(){

  led2state = (led2state == HIGH) ? LOW : HIGH;

  digitalWrite (led2, led2state);}

** **

**Task 2**

Write a sketch to implement the one-way traffic light controller using FSM concepts. The sensor will work to sense the traffic on the road whose output will be the stimulus for the state transition.

#define goRed 0

#define goYellow 1

#define goGreen 2

int red = 3;

int yellow = 4;

int green = 5;

int sen = 7;

int senState;

static unsigned long ts;

int state = goRed;

bool flag = HIGH;

void setup() {

  pinMode(red, OUTPUT);

  pinMode(yellow, OUTPUT);

  pinMode(green, OUTPUT);

  pinMode(sen, INPUT);}

void loop() {

  senState = digitalRead(sen);

  switch (senState) {

    case HIGH:

      switch (state) {

        case goGreen:

          digitalWrite(red, LOW);

          digitalWrite(yellow, LOW);

          digitalWrite(green, HIGH);

          state = goGreen;

          break;

        case goYellow:

          digitalWrite(red, LOW);

          digitalWrite(yellow, HIGH);  // When Sensor is Sensing the Traffic

          digitalWrite(green, LOW);

          state = goGreen;

          break;

        case goRed:

          digitalWrite(red, HIGH);

          digitalWrite(yellow, LOW);

          digitalWrite(green, LOW);

          state = goGreen;

          break;}

      break;

    case LOW:

      switch (state) {

        case goGreen:

          digitalWrite(red, LOW);

          digitalWrite(yellow, LOW);

          digitalWrite(green, HIGH);

          state = goYellow;

          break;

        case goYellow:

          if (flag) {

            ts = millis();}

          flag = LOW;

          digitalWrite(red, LOW);  // When Sensor is Not Sensing the Traffic.

          digitalWrite(yellow, HIGH);

          digitalWrite(green, LOW);

          if (millis() > ts + 2000) {

            state = goRed;

            flag = HIGH;}

          break;

        case goRed:

          digitalWrite(red, HIGH);

          digitalWrite(yellow, LOW);

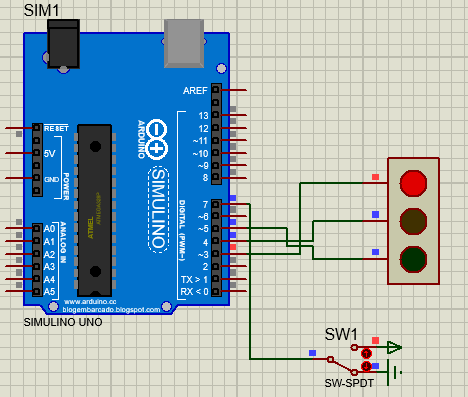
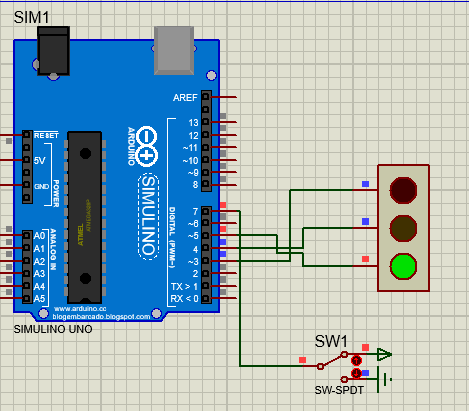
          digitalWrite(green, LOW);

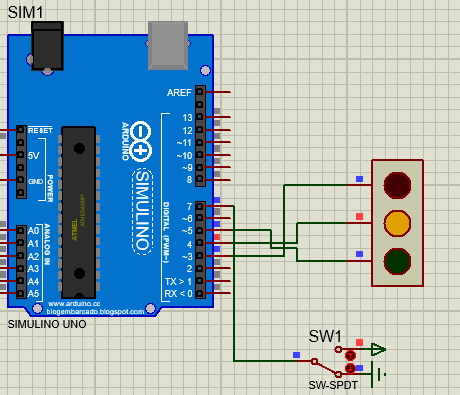
          state = goRed;

          break;}

      break;}

}

** **

****