[Materials Today Proceeding Journal]

Green Energy Powered - Vapor, Thermal and UV Light Assisted Disinfection Technology

Kishor Kumar Sadasivunia,\*, Muni Raj Mauryaa,b, Mohammad Talal Houkanb, John-John Cabibihanb, Mithra Geethaa, Somaya Al-Maadeedc, Nazreen Che Rosland, Dania Adila Ahmad Ruzaidid, Mohd Muzamir Mahatd

aCenter for Advanced Materials, Qatar University, PO Box 2713, Doha, Qatar

bDepartment of Mechanical and Industrial Engineering, Qatar University, PO Box 2713, Doha, Qatar

cDepartment of Computer Science and Engineering, Qatar University, PO Box 2713, Doha, Qatar

dSchool of Physics and Material Studies, Faculty of Applied Sciences, Universiti Teknologi MARA (UITM), Shah Alam, 40450, Selangor, Malaysia

Program

#include <LiquidCrystal\_I2C.h>

#include <Wire.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#define ONE\_WIRE\_BUS 11 // pin for temperature sensor

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

LiquidCrystal\_I2C lcd(0x27, 2, 16);

const int controlPin[12] = {1,3,5,4,6,7,8,10,12,13,9,0}; // define pins

int check=0, disinfection, set=0;

float temp;

int tempPin = 0;

int loopbreak=0;

int r=0;

void setup()

{

lcd.init();

lcd.backlight();//To Power ON the back light

lcd.clear();

delay(1000);

pinMode(controlPin[0],INPUT);//IR for desinfecting

pinMode(controlPin[1],OUTPUT);

pinMode(controlPin[2],OUTPUT);

pinMode(controlPin[3],INPUT); //IR for opening and closing

pinMode(controlPin[4],OUTPUT);

pinMode(controlPin[5],OUTPUT);

pinMode(controlPin[6],OUTPUT);

pinMode(controlPin[7],INPUT);

pinMode(controlPin[8],OUTPUT);

pinMode(controlPin[9],OUTPUT);

pinMode(controlPin[10],OUTPUT);

pinMode(controlPin[11],OUTPUT);

digitalWrite(controlPin[9], HIGH);

digitalWrite(controlPin[1], LOW);

digitalWrite(controlPin[2], LOW);

digitalWrite(controlPin[4], HIGH);

digitalWrite(controlPin[5], HIGH);

digitalWrite(controlPin[6], LOW);

digitalWrite(controlPin[7], LOW);

digitalWrite(controlPin[8], LOW);

digitalWrite(controlPin[10],LOW);

digitalWrite(controlPin[11],LOW);

sensors.begin();

Serial.begin(9600);// initialize serial monitor with 9600 baud

}

void loop()

{

if (r == 1)

{

lcd.init();

lcd.clear();

delay (1000);

r=0;

}

lcd.setCursor(0,0); //Defining positon to write from first row,first column .

lcd.print("REACTIVE-BIN");

lcd.setCursor(2, 1);

lcd.print("Hello");

delay (1000);

displaytem ();

if(digitalRead(controlPin[3])== LOW && check==0)

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Gate Opening");

displaytem ();

digitalWrite(controlPin[2], HIGH); // gate opening

delay(1000);

check=1;

disinfection=0;

lcd.clear();

r=1;

}

if(digitalRead(controlPin[3]) == LOW && check==1)

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Gate closing");

digitalWrite(controlPin[2], LOW); // gate closing

displaytem ();

delay(1000);

check=0;

disinfection=1;

lcd.clear();

r=1;

}

if (digitalRead(controlPin[0]) == LOW && disinfection==1)

{

vapour(4,15);

lcd.init();

lcd.clear();

thermal (8,5);

lcd.init();

lcd.clear();

uvlight(9,2);

lcd.init();

}

}

void vapour(int relayChannel, int t)

{

lcd.init();

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Vapour ON");

digitalWrite(controlPin[10], HIGH);// pump motar

delay (10000);

digitalWrite(controlPin[10], LOW);

delay (1000);

digitalWrite(controlPin[relayChannel], LOW);// power supply to humidifier

delay(1000);

digitalWrite(controlPin[5], LOW); // trigerring the humidifier pin

delay(1000);

digitalWrite(controlPin[5], HIGH);

for (int i=0; i<t; i++)

{

int s;

displaytem ();

lcd.setCursor(1,1);

s=t-i;

lcd.print(s);

lcd.setCursor(5,1);

lcd.print("sec");

delay(1000);

lcd.setCursor(1,1);

lcd.print(" ");

}

digitalWrite(controlPin[relayChannel], HIGH);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Vapour OFF");

delay(1000);

lcd.clear();

}

void thermal(int relayChannel, int t)

{

int z;

lcd.init();

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Thermal ON");

digitalWrite(controlPin[relayChannel], HIGH);

for (int i=0; i<t; i++)

{

int s;

// displaytem ();

lcd.setCursor(1,1);

s=t-i;

lcd.print(s);

lcd.setCursor(5,1);

lcd.print("sec");

delay(1000);

lcd.setCursor(1,1);

lcd.print(" ");

sensors.requestTemperatures();

z = sensors.getTempCByIndex(0);

if (z > 80)

{

digitalWrite(controlPin[relayChannel], LOW);

break ;

}

}

digitalWrite(controlPin[relayChannel], LOW);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Thermal OFF");

delay(1000);

lcd.clear();

}

void uvlight(int relayChannel, int t)

{

lcd.init();

lcd.clear();

lcd.setCursor(0,0);

lcd.print("UV-C ON");

digitalWrite(controlPin[relayChannel], LOW);

for (int i=0; i<t; i++)

{

int s;

displaytem ();

lcd.setCursor(1,1);

s=t-i;

lcd.print(s);

lcd.setCursor(5,1);

lcd.print("sec");

delay(1000);

lcd.setCursor(1,1);

lcd.print(" ");

}

digitalWrite(controlPin[relayChannel], HIGH);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("UV-C OFF");

delay(1000);

lcd.clear();

}

void displaytem ()

{

lcd.setCursor(12,1);

sensors.requestTemperatures();

int t = sensors.getTempCByIndex(0);

lcd.print(t);

//lcd.print(char(223));

lcd.print("C");

}

long microsecondsToInches(long microseconds)

{

return microseconds / 74 / 2;

}

long microsecondsToCentimeters(long microseconds)

{

return microseconds / 29 / 2;

}