

#include <Arduino.h>

const int button\_pin = 9;

const int led\_pin =  11;

int button\_state = 0;

int amp=127;

int timing=0;

void setup() {

pinMode(button\_pin, INPUT);

pinMode(led\_pin, OUTPUT);

}

void loop() {

button\_state = digitalRead(button\_pin);

if(button\_state == HIGH && timing < 15 && amp<=255)

{

analogWrite(led\_pin, amp);

}

  else if (button\_state == HIGH && timing>15 && amp<=255)

  {

    analogWrite(led\_pin, amp);

    amp=amp+16;

  }

  else if(amp>255 && button\_state == HIGH)

  {

    analogWrite(led\_pin, 0);

  }

  else if(button\_state == LOW)

  {

    analogWrite(led\_pin, 0);

    timing=0;

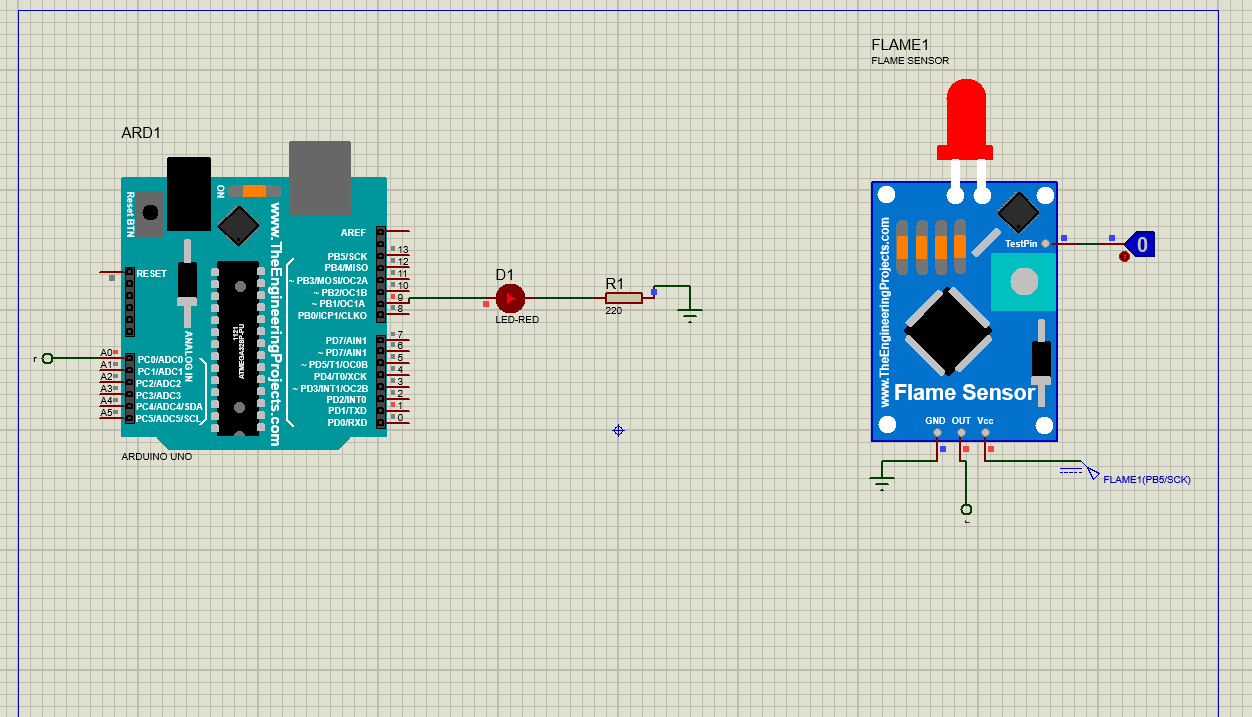
    amp=255;

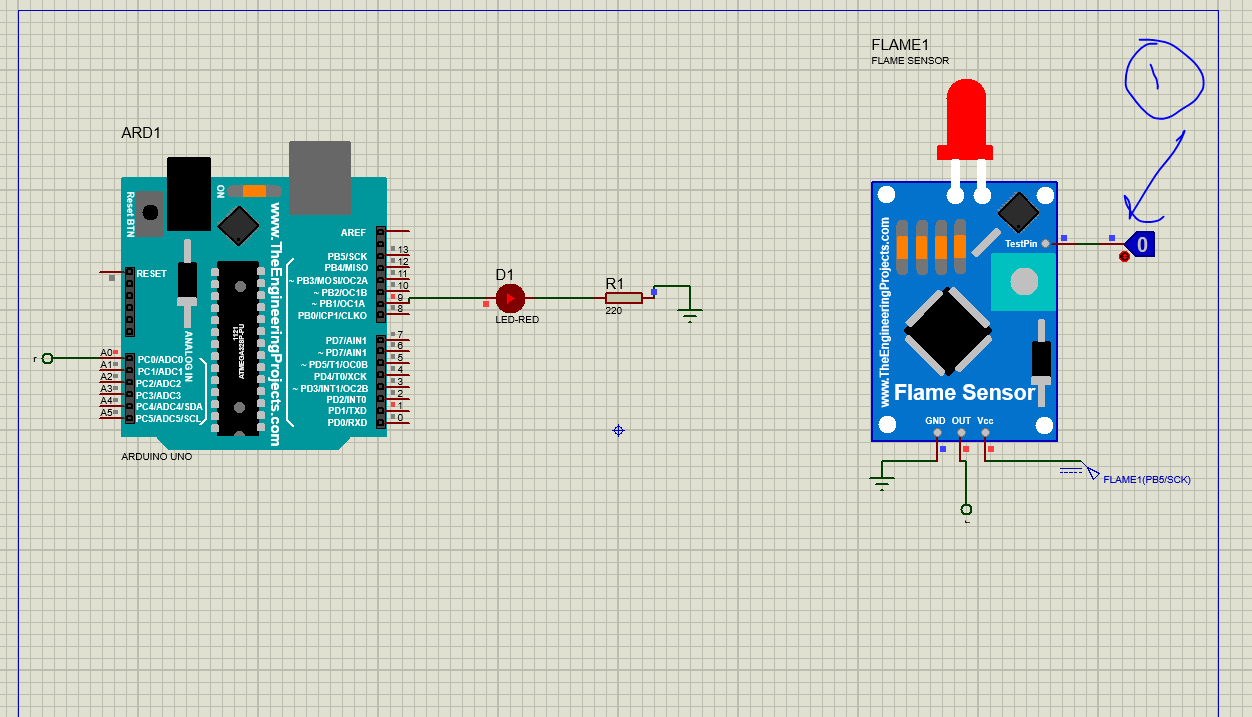
  }

  timing=timing+1;

  delay(200);

}





#include <Arduino.h>

int read\_flame = A0;

int led\_pin = 9;      // select the pin for the LED

int sensor\_value = 0;  // variable to store the value coming from the sensor

void setup() {

  // declare the led\_pin as an OUTPUT:

  pinMode(led\_pin, OUTPUT);

  pinMode(read\_flame, INPUT);

  Serial.begin(9600);

}

void loop() {

  sensor\_value = digitalRead(read\_flame);

  if(sensor\_value==HIGH)

  {

    digitalWrite(led\_pin, HIGH);

  }

  else

  {

    digitalWrite(led\_pin, LOW);

  }

}

Assignment\_2

Code for ARD\_ logic:

const int trigger = 5; // PWM trigger

const int echo=3;// PWM Output 0-25000US,Every 50US represent 1cm

long duration; // defines variables

float distance;

int input;

// the pen numbers of the leds

int RED=11;

int GREEN=10;

int BLUE=9;

int YELLOW=8;

void setup()

{

pinMode(trigger, OUTPUT); //Sets the trigger as an Output

pinMode(RED, OUTPUT);

pinMode(GREEN, OUTPUT);

pinMode(BLUE, OUTPUT);

pinMode(YELLOW, OUTPUT);

pinMode(echo, INPUT); // Sets the echopin as an Input

Serial.begin(9600); //configure baud rate of Serial communication to 9600 bps

}

void loop()

{

digitalWrite(trigger, LOW); //Clears the triggerpen

delayMicroseconds(2); // Sets the triggerpen on HIGH state for 10 micro seconds

digitalWrite(trigger, HIGH);

delayMicroseconds(10);

digitalWrite(trigger, LOW);

duration=pulseIn(echo,HIGH);// Reads the echopin, returnsthe sound wave travel time in microseconds

distance= duration\*0.034/1050.0;//i got this formula by trying to map the potentiometer readings with the distance range;

input=analogRead(A0);

Serial.print("Distance Measured="); // Prints the distance on the Serial Monitor

Serial.print(distance);

Serial.println("cm");

if (distance==2)

{

digitalWrite(RED, HIGH);

digitalWrite(GREEN, HIGH);

digitalWrite(BLUE, HIGH);

digitalWrite(YELLOW, HIGH);

Serial.println("2 Cm High");

Serial.println("1.5 Cm High");

Serial.println("1 Cm High");

Serial.println(".5 Cm High");

}

else if (distance>1.5)

{

digitalWrite(RED, LOW);

digitalWrite(GREEN, HIGH);

digitalWrite(BLUE, HIGH);

digitalWrite(YELLOW, HIGH);

Serial.println("2 Cm LOW");

Serial.println("1.5 Cm High");

Serial.println("1 Cm High");

Serial.println(".5 Cm High");

}

else if (distance>1.0)

{

digitalWrite(RED, LOW);

digitalWrite(GREEN, LOW);

digitalWrite(BLUE, HIGH);

digitalWrite(YELLOW, HIGH);

Serial.println("2 Cm LOW");

Serial.println("1.5 Cm LOW");

Serial.println("1 Cm High");

Serial.println(".5 Cm High");

}

else if (distance>.5)

{

digitalWrite(RED, LOW);

digitalWrite(GREEN, LOW);

digitalWrite(BLUE, LOW);

digitalWrite(YELLOW, HIGH);

Serial.println("2 Cm LOW");

Serial.println("1.5 Cm LOW");

Serial.println("1 Cm LOW");

Serial.println(".5 Cm High");

}

else if (distance<=.5)

{

digitalWrite(RED, LOW);

digitalWrite(GREEN, LOW);

digitalWrite(BLUE, LOW);

digitalWrite(YELLOW, LOW);

Serial.println("2 Cm LOW");

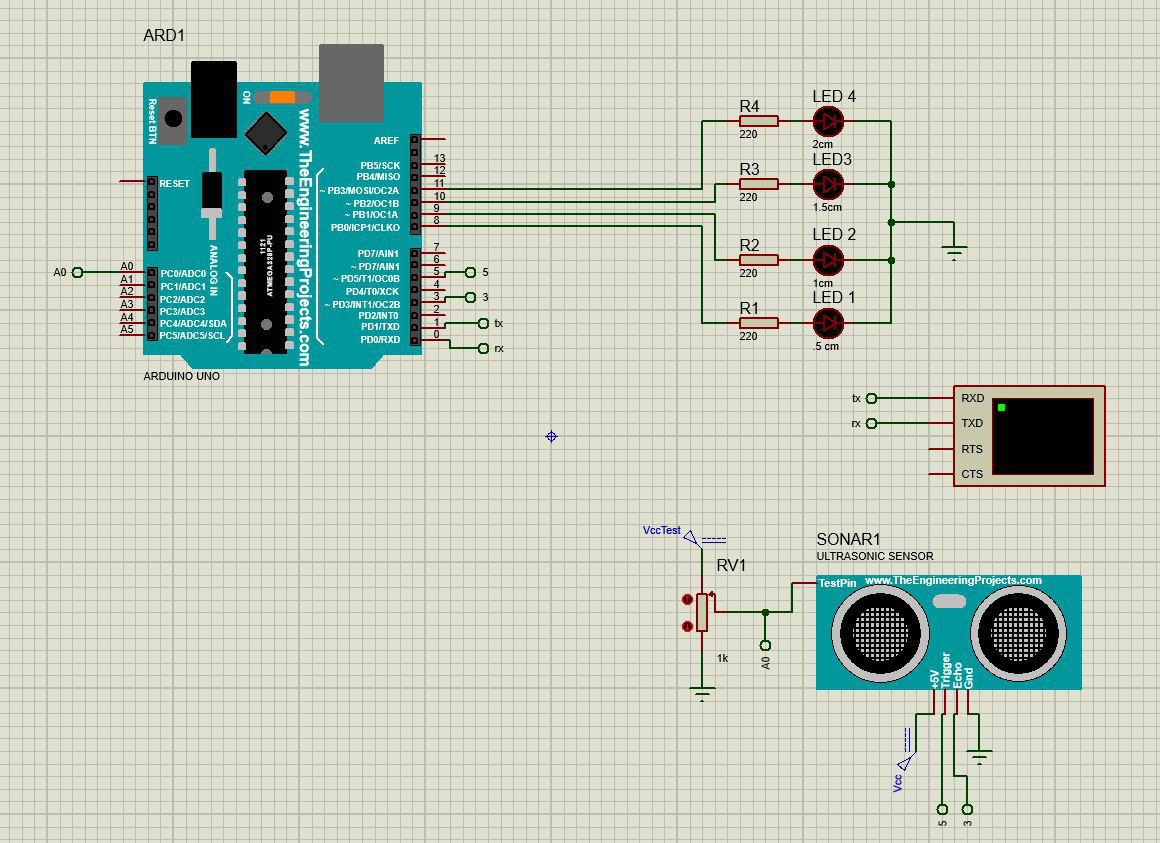
Serial.println("1.5 Cm LOW");

Serial.println("1 Cm LOW");

Serial.println(".5 Cm LOW");

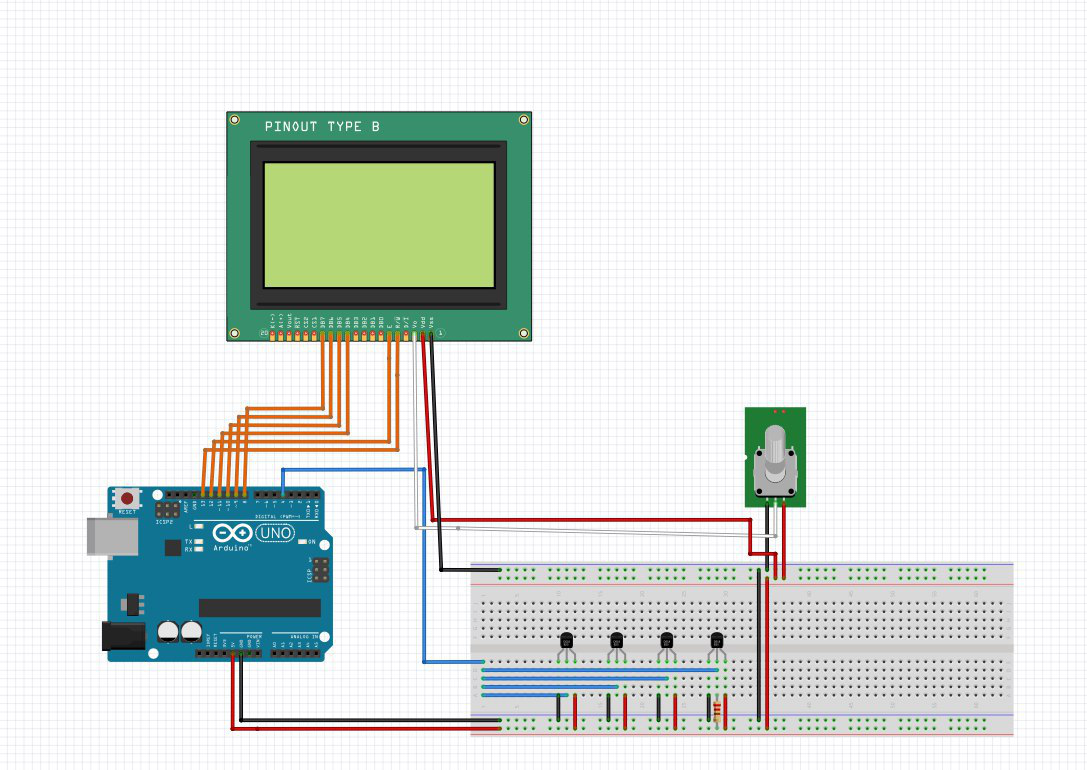
}

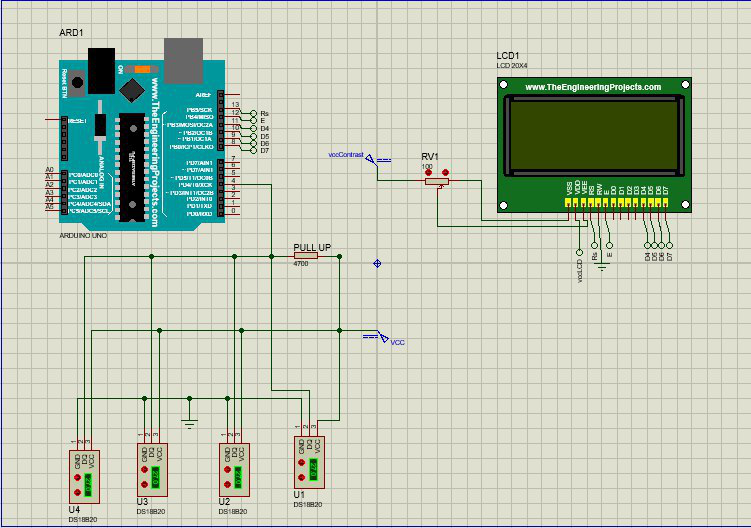
}



Assignment 3:

Pepsi:





Code:

#include <OneWire.h>

#include <DallasTemperature.h>

#include <LiquidCrystal.h>

// Data wire is plugged into port 4 on the Arduino

#define ONE\_WIRE\_BUS 4

// Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dallas temperature ICs)

OneWire oneWire(ONE\_WIRE\_BUS);

// Pass our oneWire reference to Dallas Temperature.

DallasTemperature sensors(&oneWire);

int numberOfDevices; // Number of temperature devices found

DeviceAddress tempDeviceAddress; // We'll use this variable to store a found device address

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

void setup(void) {

// Start up the library

sensors.begin();

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

lcd.setCursor(1,0);

// Grab a count of devices on the wire

numberOfDevices = sensors.getDeviceCount();

// locate devices on the bus

lcd.setCursor(1,0);

lcd.print("Locating devices...");

lcd.print("Found ");

lcd.print(numberOfDevices, DEC);

lcd.println(" devices.");

delay(1000);

// Loop through each device, print out address

for(int i=0;i<numberOfDevices; i++) {

lcd.clear();

lcd.setCursor(1,0);

// Search the wire for address

delay(50);

if(sensors.getAddress(tempDeviceAddress, i)) {

lcd.print("Found device ");

lcd.print(i, DEC);

lcd.print(" with address: ");

printAddress(tempDeviceAddress);

lcd.println();

} else {

lcd.print("Found ghost device at ");

lcd.print(i, DEC);

lcd.print(" but could not detect address. Check power and cabling");

}

}

}

void loop(void) {

lcd.clear();

lcd.setCursor(1,0);

sensors.requestTemperatures(); // Send the command to get temperatures

// Loop through each device, print out temperature data

for(int i=0;i<numberOfDevices; i++) {

// Search the wire for address

if(sensors.getAddress(tempDeviceAddress, i)){

// Output the device ID

lcd.print("Temperature for device: ");

lcd.println(i,DEC);

// Print the data

float tempC = sensors.getTempC(tempDeviceAddress);

lcd.print("Temp C: ");

lcd.print(tempC);

lcd.print(" Temp F: ");

lcd.println(DallasTemperature::toFahrenheit(tempC)); // Converts tempC to Fahrenheit

}

}

delay(5000);

}

// function to print a device address

void printAddress(DeviceAddress deviceAddress) {

for (uint8\_t i = 0; i < 8; i++) {

if (deviceAddress[i] < 16) lcd.print("0");

lcd.print(deviceAddress[i], HEX);

}

}