Project Development Phase
Performance Test

Date	20 May 2023
Team ID	NM2023TMID17372
Project Name	Industrial Workers Health And Safety System Based On Internet Of Things

Model Performance Testing:

Project team shall fill the following information in the performance testing template

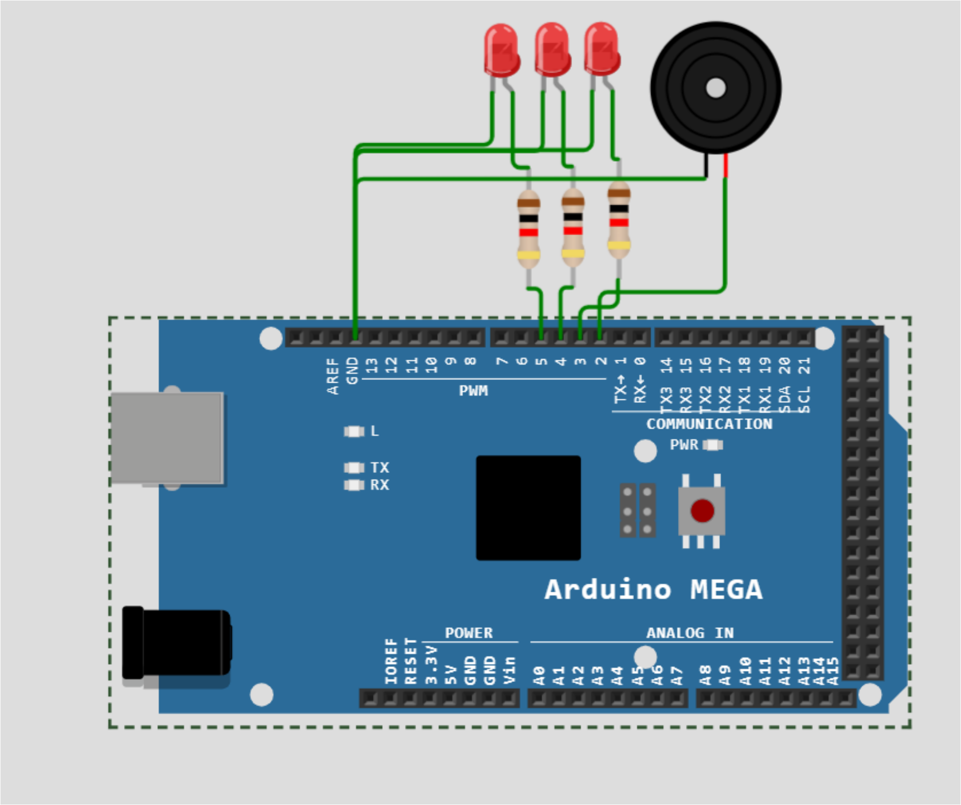
C

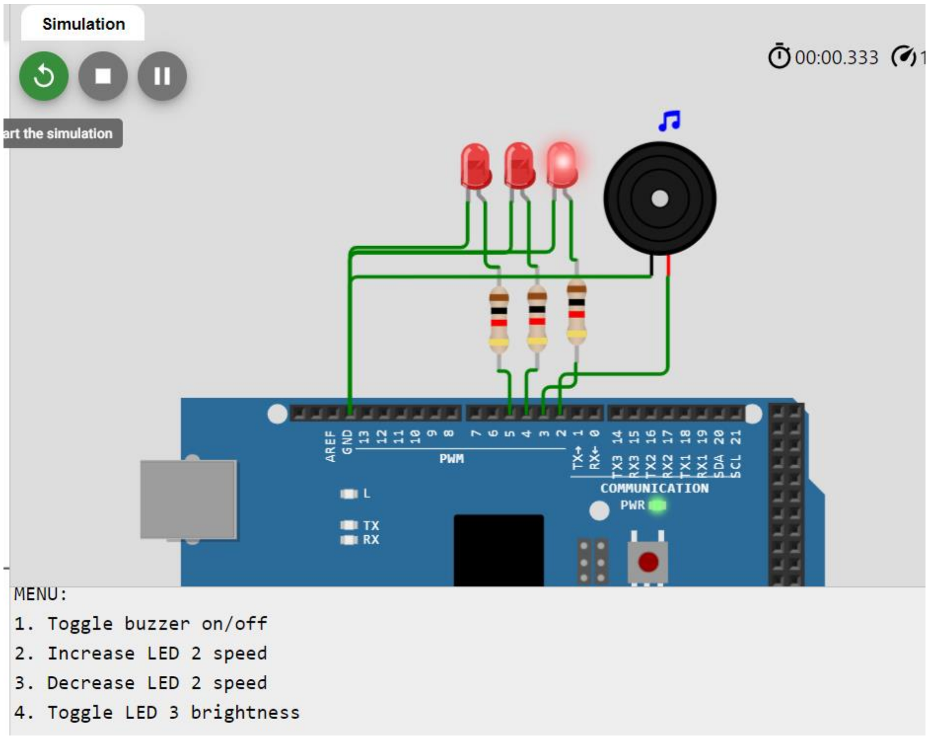
Parameter	Values	Screenshot
Metrics	Wowki Execution time and Output screenshot Or Python accuracy of prediction and output screenshot	

Coding:

```
const int ledPin1 = 3;

const int ledPin2 = 4;
const int ledPin3 = 5;
const int buzzerPin = 2;

int menuSelection = 0;
int ledSpeed = 500;
int ledBrightness = 128;
int selection = 0;
int buzzerState = LOW;

void setup() {
  Serial.begin(9600);

  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  digitalWrite(buzzerPin, LOW);
  digitalWrite(ledPin1, LOW);
  digitalWrite(ledPin2, LOW);
  digitalWrite(ledPin3, LOW);
  Serial.println("MENU:");
  Serial.println("1. Toggle buzzer on/off");
  Serial.println("2. Increase LED 2 speed");
  Serial.println("3. Decrease LED 2 speed");
  Serial.println("4. Toggle LED 3 brightness");
  Serial.println();
  Serial.print("Selection: ");
}

void loop() {
  int buzzerPinStateLast = digitalRead(buzzerPin);
  if (Serial.available()) {
    int inputChar = Serial.parseInt();

    switch (inputChar) {
      case 1:
      //Serial.println ("1");
      //digitalWrite(buzzerPin, !digitalRead(buzzerPin));
        ToggleBuzzer();
        selection = 0;
        break;
      case 2:
      Serial.println("case 2");
        ledSpeed -= 50;
        if (ledSpeed < 50) {
          ledSpeed = 50;
        }
        break;
      case 3:
      Serial.println("case 3");
        ledSpeed += 50;
        if (ledSpeed > 1000) {
          ledSpeed = 1000;
        }
        break;
      case 4:
      Serial.println("case 4");
        if (ledBrightness == 0) {
          ledBrightness = 128;
        } else {
          ledBrightness = 0;
        }
        break;
      default:
        break;
    }
  }

  digitalWrite(ledPin1, !digitalRead(ledPin1));
  delay(500);

  static unsigned long lastBlinkTime = 0;
  if (millis() - lastBlinkTime > ledSpeed) {
    digitalWrite(ledPin2, !digitalRead(ledPin2));
    lastBlinkTime = millis();
  }

  analogWrite(ledPin3, ledBrightness);
//Serial.println("MENU:");
  //Serial.println("1. Toggle buzzer on/off");
  //Serial.println("2. Increase LED 2 speed");
  //Serial.println("3. Decrease LED 2 speed");
  //Serial.println("4. Toggle LED 3 brightness");
  //Serial.println();
  //Serial.print("Selection: ");
  //delay (5000)

}
void ToggleBuzzer ()
{
  buzzerState= (buzzerState) ? LOW : HIGH;
    digitalWrite(buzzerPin, buzzerState);
  //int a = digitalWrite(buzzerPin, LOW);
  //if (a == 1)
  //{
    //digitalWrite(buzzerPin, HIGH);
    //digitalWrite(buzzerPin HIGH); attempt no. 3 failed with multiple errors
 // } else
 // {
 //   digitalWrite(buzzerPin, LOW);
 // }

}
```

Output:


AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TX→ 1
RX← 0
14
15
16
17
18
19
20
21
TX3
RX3
TX2
RX2
TX1
RX1
SDA
SCL
PWM
COMMUNICATION
L
TX
RX
PWR
Arduino MEGA
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15



```
MENU:
1. Toggle buzzer on/off
2. Increase LED 2 speed
3. Decrease LED 2 speed
4. Toggle LED 3 brightness
```