**Assignment-1**

**DOMAIN:** IOT

**TOPIC:** SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

**NAME:** MUKHESH KARAN B.L

**REG NUMBER:** 727819TUCS131

**COLLEGE:** SRI KRISHNA COLLEGE OF TECHNOLOGY

**CIRCUIT:**



**Components:**

|  |  |
| --- | --- |
| **Quantity** | **Components** |
| 1 | Push Button |
| 1 | Red LED |
| 1 | Blue LED |
| 1 | PIEZO Buzzer |
| 1 | Ultrasonic Distance Sensor |
| 2 | PIR Sensor |
| 2 | Resistor (220,560,10K) |
| 1 | Arduino R3 |
| 1 | Breadboard Small |

**Code:**

const int trigPin = 2;

const int echoPin = 4;

const int pirPin = 7; //the PIR sensor pin

int pirState = LOW; //basically means that the PIR sensor starts as low and detects no motion

const int buzzerPin = 8; //the buzzer has been connected to pin 8

const int redLED = 9; //the red LED; intensity can be controlled to change the colour emitted

int redBright = 0; // how bright the LED is

int redFade = 5; // how many points to fade the LED by

const int greenLED = 10; //the green LED; intensity can be controlled to change the colour emitted

int greenBright = 0; // how bright the LED is

int greenFade = 5; // how many points to fade the LED by

const int button = 13; //button to momentarily reset all the sensors back to normal

void setup() {

pinMode(echoPin, INPUT);

pinMode(pirPin, INPUT);

pinMode(button, INPUT);

pinMode(trigPin, OUTPUT);

pinMode(redLED, OUTPUT);

pinMode(greenLED, OUTPUT);

pinMode(buzzerPin, OUTPUT);

Serial.begin(9600); // initialize serial communication at 9600 bits per second

}

void distance() {

long durationInDigit;

long distanceInInches;

digitalWrite (trigPin, LOW); //set this to LOW to start with

delayMicroseconds(2); //delay in microseconds between different commands

digitalWrite (trigPin, HIGH); //here, the trig pin sends signals or vibrations to be detected

delayMicroseconds(10);

digitalWrite (trigPin, LOW); //set the the trig pin back to low

durationInDigit = pulseIn(echoPin, HIGH);

distanceInInches = durationInDigit/74/2;

Serial.println(distanceInInches);

if (distanceInInches > 15 && distanceInInches < 30) {

digitalWrite(greenLED, HIGH);

digitalWrite(redLED, LOW);

}

if (distanceInInches < 10) {

digitalWrite(redLED, HIGH);

digitalWrite(greenLED, LOW);

}

if (distanceInInches > 10 && distanceInInches < 15){

digitalWrite(redLED, LOW);

digitalWrite(greenLED, LOW);

}

if (distanceInInches < 5) {

digitalWrite(redLED, HIGH);

tone(8, 250, 2000);

digitalWrite(greenLED, 0);

}

if (distanceInInches > 5 && distanceInInches < 10){

digitalWrite(redLED, HIGH);

digitalWrite(buzzerPin, 0);

digitalWrite(greenLED, 0);

}

if (distanceInInches > 30 || distanceInInches < 0){

Serial.println("Distance Incalculable");

}

delay(500);

}

void reset() {

if (digitalRead(button), HIGH);

digitalWrite(pirState, LOW);

digitalWrite(redLED, LOW);

digitalWrite(greenLED, HIGH);

digitalWrite(buzzerPin, 0);

//digitalWrite(echoPin, 0);

}

void loop() {

distance();

int pirState = digitalRead(pirPin);

if (pirState==1) {

Serial.println("Motion Detected!!!");

digitalWrite(greenLED, LOW);

digitalWrite(redLED, HIGH);

digitalWrite(buzzerPin, 1);

delay(500);

}

if (pirState==0) {

Serial.println("Detecting...");

digitalWrite(greenLED, HIGH);

digitalWrite(redLED, LOW);

digitalWrite(buzzerPin, 0);

delay(500);

}

}