**IIOT\_1**

#include <DHT.h> // Include the DHT library

#define dhtpin 2

#define dhttype DHT22

#define buzzerPin 3 // Define the buzzer pin

DHT dht(dhtpin, dhttype);

void setup()

{

Serial.begin(9600);

dht.begin();

pinMode(buzzerPin, OUTPUT); // Set buzzer pin as output

}

void loop()

{

float temperature dht.readTemperature();

if (temperature > 30)

{

sendAlert("Temperature Alert!l");

digitalWrite(buzzerPin, HIGH);

delay(2000); // Buzzer sound duration (1 second)

digitalWrite(buzzerPin, LOW); // Turn off the buzzer

delay(2000);

}

void sendAlert(String msg)

{

Serial.println(msg);

}

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

**IIOT\_2**

const int PIR\_SENSOR\_OUTPUT\_PIN = 4;

int warm\_up;

// Initialize warm\_up to 0

void setup()

{

pinMode (PIR\_SENSOR\_OUTPUT\_PIN, INPUT);

Serial.begin(9600); delay(20000);

// Allow time for the sensor to warm up

}

void loop()

{

int sensor\_output;

sensor\_output = digitalRead(PIR\_SENSOR\_OUTPUT\_PIN);

if (sensor\_output == LOW)

// No object detected

{

if (warm\_up == 1) {

Serial.print("Warming up\n\n");

// Print warming up message

warm\_up = 0;

// Reset warm\_up to e

delay(20000);

// Delay for 20 seconds

}

Serial.print("NO object detected\n\n");

// Print no object detected message

delay(1000);

// Delay for 1 second

} else {

Serial.print("Object Detected\n\n");

warm\_up = 1;

// Set warm\_up to 1

// Print object detected message.

delay(1000);

// Delay for 1 second

}

}

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

**IIOT\_3**

const int ledPin = 5; // LED connected to digital pin 5

const int 1drPin = A0;

void setup()

{

Serial.begin(9600);

// LDR connected to analog pin A0 (assuming Ao as an example

// Start serial communication at 9600 baud rate

pinMode(ledPin, OUTPUT); // Set ledPin as an OUTPUT pinMode(1drPin, INPUT); // Set 1drPin as an INPUT

}

void loop()

{

int ldrstatus = analogRead(IdrPin); // Read the LDR value

if (ldrStatus <= 320)

// If it's dark

digitalWrite(ledPin, HIGH); // Turn on the LED

Serial.print("Darkness over here, turn on LED. LDR Value: ");

Serial.println(ldrStatus);

{

}

else

// If there's sufficient light

(

digitalWrite(ledPin, LOW); // Turn off the LED

Serial.print("There is sufficient light, turn off the LED. LDR Value: ");

Serial.println(ldrStatus);

}

delay(1000); // Wait for 1 second before the next loop

}

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

**IIOT\_4**

int buzzPin = 7;

// Buzzer connected to pin 7

int trigPin = 6;

// Trigger pin for ultrasonic sensor

int echoPin = A0;

// Echo pin for ultrasonic sensor

int IRPin A1;

// IR sensor connected to analog pin A1

int IRV;

// Variable to store IR sensor value

int duration, distance;

// Variables for distance measurement

void setup()

{

pinMode(trigPin, OUTPUT);

// Set trigPin as an OUTPUT

pinMode(buzzPin, OUTPUT);

// Set buzzpin as an OUTPUT

pinMode(echoPin, INPUT);

// Set echopin as an INPUT

pinMode (IRPin, INPUT);

// Set IRPin as an INPUT

Serial.begin(9600);

// Start serial communication at 9600 baud rate

}

void loop()

{

// Measure distance using ultrasonic sensor

digitalWrite(trigPin, LOW);

delayMicroseconds (2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration= pulseIn(echoPin, HIGH); // Read the pulse duration distance = duration 0.034 / 2; // Calculate distance in cm

Serial.print("Distance in CM is: ");

Serial.println(distance);

// Check if an obstacle is detected

if (distance < 20) {

digitalWrite(buzzPin, HIGH); // Turn on the buzzer

Serial.println("Obstacle detected");

} else {

digitalWrite(buzzPin, LOW); // Turn off the buzzer

Serial.println("Obstacle not detected");

}

delay(100); // Wait for 100 ms before the next measurement

// Read the value from the IR sensor

IRV digitalRead(IRPin);

Serial.print("IR Sensor Value: ");

Serial.println(IRV);

delay(1000); // Wait for 1 second before the next loop

}

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

**IIOT\_5**

const int buzzerPin = 5;

// Buzzer connected to digital pin 5

const int flamePin = 2;

// Flame sensor connected to digital pin 2

void setup() {

pinMode(buzzerPin, OUTPUT);

// Set buzzer pin as output

pinMode(flamePin, INPUT);

// Set flame sensor pin as input

Serial.begin(9600);

// Start serial communication

}

void loop() {

int flame= digitalRead(flamePin);

// Read the flame sensor value

if (flame == LOW) {

// Check if flame is detected

Serial.println("Fire is Detected");

digitalWrite(buzzerPin, HIGH);

// Turn on the buzzer

} else {

Serial.println("No Fire is Detected");

digitalWrite(buzzerPin, LOW);

// Turn off the buzzer

}

delay(1000); // Wait for 1 second before the next loop iteration

}

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