// Define the sound sensor pin

const int soundSensorPin = A0;

// Define the buzzer pin

const int buzzerPin = 9;

// Define the reset button pin

const int resetButtonPin = 2;

// Define the threshold for sound detection

const int soundThreshold = 500;

// Define variables for sound detection and alarm state

int soundValue = 0;

bool alarmActive = false;

void setup() {

// Set the sound sensor pin as an input

pinMode(soundSensorPin, INPUT);

// Set the buzzer pin as an output

pinMode(buzzerPin, OUTPUT);

// Set the reset button pin as an input with internal pull-up resistor enabled

pinMode(resetButtonPin, INPUT\_PULLUP);

// Initialize the buzzer to an off state

digitalWrite(buzzerPin, LOW);

}

void loop() {

// Read the sound sensor value

soundValue = analogRead(soundSensorPin);

// Check if the sound value exceeds the threshold and the alarm is not active

if (soundValue > soundThreshold && !alarmActive) {

// Activate the alarm

activateAlarm();

}

if (digitalRead(resetButtonPin) == HIGH && alarmActive) {

// Deactivate the alarm

deactivateAlarm();

}

}

void activateAlarm() {

// Sound the alarm by turning on the buzzer

digitalWrite(buzzerPin, HIGH);

alarmActive = true;

}

void deactivateAlarm() {

// Silence the alarm by turning off the buzzer

digitalWrite(buzzerPin, LOW);

alarmActive = false;

// Wait until the reset button is released

while (digitalRead(resetButtonPin) == LOW) {

delay(10);

}

}