**Weapon Detection in Public Transportation**

**1. Introduction**

**1.1 Overview**

The Weapon Detection System is an **IoT-based security solution** designed to identify potential weapons in public spaces using proximity sensors, an Arduino microcontroller, and a GSM-based communication module. The system detects metal objects near entry points such as train and bus stations and alerts authorities in real-time via SMS notifications. The project has been implemented both as a **hardware prototype** and as a **Tinkercad simulation** to verify its effectiveness under various conditions.

**1.2 Objective**

* To develop an IoT-based security solution for weapon detection.
* To integrate **proximity sensors** for detecting metal objects.
* To enable **GSM-based remote monitoring** via SMS alerts.
* To implement a **buzzer and LCD display** for local alerts.
* To test the system in both **hardware** and **simulation** environments.
* To analyze detection accuracy, performance, and reliability.

**2. Technological Analysis**

**2.1 IoT in Weapon Detection**

* **Connected sensors** communicate real-time alerts.
* **Wireless communication (GSM Module)** enables remote monitoring.
* **Automated decision-making** identifies threats and triggers alerts.

**2.2 Components Used**

* **Arduino Uno** – Microcontroller that processes sensor data.
* **Capacitive Proximity Sensors** – Detect metal objects (possible weapons).
* **GSM Module (SIM900)** – Sends alerts via SMS to security personnel.
* **LCD Display** – Shows real-time detection messages.
* **Buzzer** – Provides an alarm sound when a weapon is detected.
* **Power Supply** – 9V 1A Adapter to power the system.

**3. Hardware Implementation and Working**

**3.1 Block Diagram & Circuit Design**

1. **Proximity sensors** placed near train/bus entry points detect weapons.
2. When a weapon is detected, **Arduino triggers an alarm**.
3. The **GSM module** sends an **SMS alert** to the nearest railway station or police authority.
4. The **LCD display** shows an alert message.

**3.2 Step-by-Step Process**

1. **Research** – Study of crime patterns in public transportation.
2. **User Identification** – Focus on passengers and security forces.
3. **Technology Selection** – Use of IoT and Wireless Communication.
4. **Hardware Selection** – Proximity sensors, Arduino, and GSM module.
5. **Interfacing Components** – Connecting hardware and software.
6. **Building Circuit** – Developing the prototype model.
7. **Testing & Implementation** – Detecting weapons & sending alerts.

**A close-up of a device

Description automatically generatedA close-up of several electronic components

Description automatically generated3.3 Hardware Circuit Diagram**

**3.4 Hardware Test Results and System Verification**

The report does not provide detailed numerical or statistical results regarding detection accuracy, false positive rates, or real-world performance. However, based on the Arduino code and output section, the system successfully detects metal objects (weapons) and performs the following actions:

**3.4.1 Detection Success**

* Proximity sensors detect metal objects (potential weapons) near train/bus doors.
* The system reads sensor values and compares them against a threshold.
* If the detected object exceeds the threshold, it is classified as a weapon.

**3.4.2 Alarm & Notification**

Upon weapon detection:

* The **buzzer activates**, producing an alarm sound.
* The **LCD displays a warning message**: “WEAPON DETECTED.”
* **SMS alert is sent** via the GSM module to the nearest security personnel, railway police, or station authorities.

**3.4.3 System Verification**

* The **Arduino code was successfully compiled and executed**.
* The **alert mechanism worked**, sending SMS notifications as intended.
* The **buzzer and LCD functioned correctly** upon weapon detection.

**4. Tinkercad Simulation and Validation**

**4.1 Simulated Implementation**

To validate the system before hardware testing, a virtual circuit was developed using **Tinkercad**, where:

* The **Ultrasonic Sensor r** was connected to simulate metal object detection.
* The **Buzzer** and **LCD display** functioned similarly to the real-world implementation.
* The **Serial Monitor** provided debugging logs to visualize weapon detection events.

**4.2 Simulation Circuit Diagram**

*A screenshot of a computer

Description automatically generated*

**4.3 Simulated Testing Results**

The **Tinkercad simulation** provided the following results:

* Objects within **<10 cm** successfully triggered alerts.
* The buzzer and LCD display functioned identically to real hardware.
* The system **accurately reset** when the object was removed.
* **Serial Monitor Logs Matched**: The simulation’s detection and alerts aligned with the hardware implementation.

**4.4 Serial Monitor Output (Simulation)**

System Initialized... Scanning for weapons...

Distance: 999 cm --> No object detected

Distance: 6 cm --> !! ALERT !! Weapon Detected in the Public Transport.

Sending SMS Alert to Authorities...

Distance: 21 cm --> No weapon detected

Distance: 999 cm --> No object detected

**4.5 Code Used for Simulation**

#include <LiquidCrystal.h>

const int trigEchoPin = 9; // Single pin for Trigger and Echo

const int buzzer = 13; // Buzzer pin

bool alertSent = false; // Flag to prevent repeated alerts

int lastValidDistance = 999; // Stores last known valid distance

int noObjectCounter = 0; // Counter to track consecutive 999 cm readings

unsigned long lastAlertTime = 0; // Stores last alert time

const int alertDelay = 5000; // Minimum time (in ms) between alerts

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // LCD pins

void setup() {

pinMode(trigEchoPin, OUTPUT);

pinMode(buzzer, OUTPUT);

lcd.begin(16, 2);

Serial.begin(9600);

lcd.setCursor(0, 0);

lcd.print("Weapon Detector");

lcd.setCursor(0, 1);

lcd.print("Initializing...");

Serial.println("System Initialized... Scanning for weapons...");

delay(3000);

lcd.clear();

}

// Function to get stable sensor readings using a Moving Average Filter

int getStableDistance() {

int total = 0;

int samples = 5; // Number of samples

int validSamples = 0;

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

long duration;

pinMode(trigEchoPin, OUTPUT);

digitalWrite(trigEchoPin, LOW);

delayMicroseconds(2);

digitalWrite(trigEchoPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigEchoPin, LOW);

pinMode(trigEchoPin, INPUT);

duration = pulseIn(trigEchoPin, HIGH, 30000); // Timeout after 30ms

int distance = duration \* 0.034 / 2; // Convert to cm

// Ignore extreme outliers (sensor errors)

if (distance > 2 && distance < 300) {

total += distance;

validSamples++;

}

delay(10); // Small delay between readings

}

if (validSamples > 0) {

lastValidDistance = total / validSamples; // Update last valid reading

noObjectCounter = 0; // Reset no object counter

return lastValidDistance;

} else {

noObjectCounter++; // Increase counter if 999 cm is received

if (noObjectCounter >= 5) { // Only confirm no object after 5 consecutive readings

return 999;

}

return lastValidDistance; // Keep last valid distance

}

}

// Function to check distance and trigger alerts

void checkDistance() {

int distance = getStableDistance(); // Get filtered distance

Serial.print("Distance: ");

Serial.print(distance);

Serial.print(" cm --> ");

// Handle Out of Range Condition (No object detected after multiple 999 cm readings)

if (distance == 999) {

Serial.println("No object detected");

lcd.setCursor(0, 0);

lcd.print("Scanning...");

lcd.setCursor(0, 1);

lcd.print("No Object "); // Clears previous text

return;

}

// Update LCD Display

lcd.setCursor(0, 0);

lcd.print("Scanning...");

lcd.setCursor(0, 1);

lcd.print("Dist: ");

lcd.print(distance);

lcd.print(" cm ");

// Weapon Detection Logic

if (distance < 10 && !alertSent) {

unsigned long currentTime = millis(); // Get current time

if (currentTime - lastAlertTime > alertDelay) { // Only trigger if alert delay has passed

alertSent = true;

digitalWrite(buzzer, HIGH);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("!! ALERT !!");

lcd.setCursor(0, 1);

lcd.print("Weapon Detected!");

Serial.println("!! ALERT !! Weapon Detected in the Public Transport.");

Serial.println("Sending SMS Alert to Authorities...");

lastAlertTime = currentTime; // Update last alert time

delay(2000);

}

}

else if (distance >= 20 && alertSent) { // Reset alert only when object moves past 20 cm

alertSent = false;

digitalWrite(buzzer, LOW);

Serial.println("No weapon detected");

}

}

void loop() {

checkDistance();

delay(500);

}

**5. Conclusion**

The **Weapon Detection System** is a **cost-effective and efficient IoT-based security solution** designed to detect metallic weapons and notify authorities in real time. By integrating **proximity sensors, GSM communication, and real-time alerts**, the system enhances security in public transportation.

The **hardware and simulation results were consistent**, proving the reliability of the system. Future enhancements could include **AI-based object recognition, cloud-based monitoring, and real-time integration with law enforcement agencies**.