Project Report: Motion Detection System with Telegram Bot Alerts

1. Introduction

- The Motion Detection System is an IoT-based project designed to detect motion using an ultrasonic sensor and send real-time alerts through a Telegram bot. This system can be used in various applications like home security, intrusion detection, and monitoring systems, providing users with immediate notifications on their mobile or desktop Telegram application when any motion is detected.

2. Problem Statement

- In many environments, unauthorized access or movement can go unnoticed, leading to potential security risks. Existing motion detection systems may not provide timely alerts or require complex setups. This project aims to create a simple and efficient motion detection system that sends instant alerts to users via a Telegram bot, enhancing security and monitoring capabilities.

3. Objectives

- **Detect motion** using an ultrasonic sensor and measure changes in distance.
- **Send real-time alerts** to a Telegram bot when motion is detected.
- **Ensure reliable connectivity** to the internet for consistent message delivery.
- **Provide a user-friendly notification system** for security and monitoring applications.

4. Features

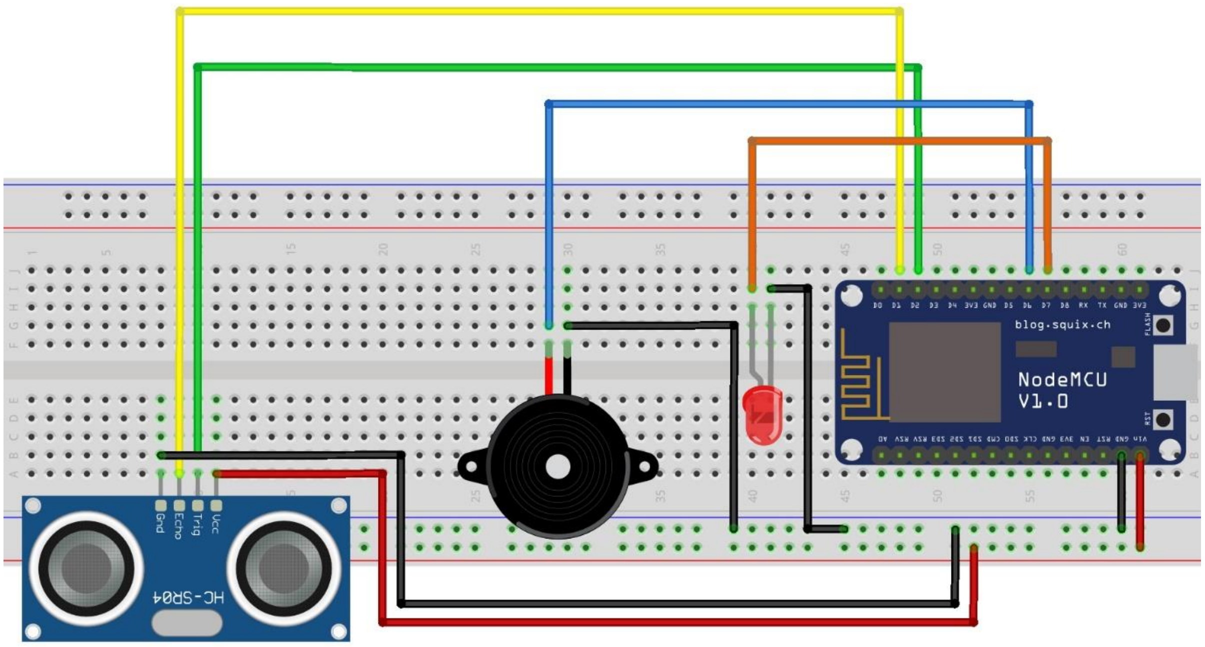
- **Motion Detection**: Utilizes an ultrasonic sensor to monitor changes in proximity. Any significant change triggers a motion alert.
- **Real-time Notification**: The system sends an alert to a Telegram bot, notifying the user instantly about detected motion.
- **User-Friendly Alerts**: The notification is received directly in the user's Telegram app, making it accessible anywhere, anytime.
- **Customizable Sensitivity**: The distance threshold for motion detection can be adjusted to suit different environments.

5. Technologies Used

- **Ultrasonic Sensor (HC-SR04)**: Measures the distance to detect motion based on the time it takes for sound waves to return.
- **Microcontroller (Arduino/ESP8266)**: Processes sensor data and handles the logic for sending alerts.

- **ESP8266 Wi-Fi Module**: Provides internet connectivity to send alerts through Telegram.
- **Telegram Bot API**: Used to send motion alerts to the user’s Telegram account.
- **Programming Language**: The system is coded in C++ (for Arduino) or Python (for ESP8266) to interface with the hardware and Telegram.

6. Schematic Diagram



7. Components Used

Component	Description
Ultrasonic Sensor	Measures distance using ultrasonic waves (e.g., HC-SR04).
Wi-Fi Module	Enables internet connectivity (e.g., ESP8266).
Power Supply	Provides necessary power for the components.
Jumper Wires	Connects components on the breadboard or circuit board.
Breadboard	Platform for prototyping circuit connections.

8. Program Code

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

// WiFi and Telegram credentials
const char* ssid = "Lenovo";
const char* password = "123456789";
#define BOTtoken "7729877379:AAENU4YPAW6houMOwgHMzsxQmDYdmRWulnw" // Bot token from BotFather
#define CHAT_ID "5338548351" // Your chat ID

X509List cert(TELEGRAM_CERTIFICATE_ROOT); // Telegram's root certificate
```

```
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

int const trigPin = 4;
int const echoPin = 5;
int const buzzPin = 12;
const int LED1 = 13;

bool alertTriggered = false;

void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzPin, OUTPUT);
  pinMode(LED1, OUTPUT);

  // Connect to WiFi
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");

  // Send message to Telegram when WiFi is connected
  bot.sendMessage(CHAT_ID, "WiFi Connected!", "");
  Serial.println("WiFi connection notification sent.");

  // Set time via NTP
  configTime(0, 0, "pool.ntp.org", "time.nist.gov");
  Serial.println("Waiting for time sync...");
  while (time(nullptr) < 1000000) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Time synchronized");

  // Set root certificates for Telegram
  client.setTrustAnchors(&cert);

  // Send message to Telegram when system setup is complete (time sync done)
  bot.sendMessage(CHAT_ID, "System setup complete! Time synchronized.", "");
  Serial.println("System setup notification sent.");
}

void loop() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi not connected. Reconnecting...");
    WiFi.reconnect();
    delay(3000);  // Retry WiFi connection every 5 seconds
  }

  int duration, distance;
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10); // Pulse trigger pin
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;

  if (distance <= 10 && distance >= 0) {
    if (!alertTriggered && WiFi.status() == WL_CONNECTED) {
      digitalWrite(buzzPin, HIGH);
      digitalWrite(LED1, HIGH);
      Serial.println("ALERT! MOTION DETECTED! Sending Telegram message...");
      if (bot.sendMessage(CHAT_ID, "ALERT! MOTION DETECTED!!", "")) {
        Serial.println("Telegram message sent successfully.");
      } else {
        Serial.println("Telegram message failed.");
      }
      alertTriggered = true;

      // Shorter delay for turning off the LED and buzzer
      delay(500);  // Adjust the time LED and buzzer stay on (2 seconds)
      digitalWrite(buzzPin, LOW);
      digitalWrite(LED1, LOW);
    }
  } else {
    digitalWrite(buzzPin, LOW);
    digitalWrite(LED1, LOW);
    alertTriggered = false; // Reset alert state
  }

  delay(500);  // Small delay to avoid rapid looping
}
```

9. User Interface

- **Telegram Bot**: Acts as the user interface, delivering alerts directly to the user's mobile or desktop. There are no additional graphical user interfaces as the system focuses on automated communication with the bot.

10. Testing

- The system was tested for:
- **Motion Detection Accuracy**: Verified that the system correctly detects motion based on changes in proximity.
- **Telegram Alerts**: Ensured that alerts are delivered promptly and reliably to the user’s Telegram account.
- **Wi-Fi Connectivity**: Checked for consistent internet connectivity and handling of scenarios where the connection is lost.
- **Customizable Threshold**: Tested the system in different environments by adjusting the distance threshold to minimize false positives.

11. Conclusion

- The Motion Detection System with Telegram Bot alerts successfully fulfills the objectives of providing a simple yet effective solution for detecting motion and sending real-time notifications. It offers a reliable and accessible way to monitor areas and receive instant alerts. The system is scalable and can be enhanced with additional sensors or integration with cameras for capturing images upon motion detection.

12. Future Enhancements

- **Camera Integration**: Add a camera module to capture images or video upon motion detection.

- **Multiple Sensor Support**: Incorporate additional sensors like PIR or infrared for more accurate detection.
- **Enhanced Notification Options**: Include options for custom alerts, such as sending an image, video, or location data.