A Report

ON

Project Based Learning

"Intrusion Detection System (IDS)"

BY

Pushpak Mahajan-1032232536

Samarth Jade -1032232614

Shantanu Raje -1032233252

SY B. Tech - Electronics & Communication Engineering (Core)

Subject Teacher: Dr. Ujjwala Rawandale

SCHOOL OF ELECTRONICS & COMMUNICATION ENGINEERING

MIT-WPU

PUNE-411038

Academic Year 2024-2025

Index

1. Introduction

Ensuring the security of our homes is more important than ever, and Intrusion Detection Systems (IDS) play a vital role in keeping us safe. In this project, we'll create a smart IDS using an ESP32 microcontroller and a PIR sensor to detect motion. Additionally, we'll include a buzzer and LED indicators to alert us of any unauthorized entry.

What sets this system apart is its integration with the Blynk application, allowing you to monitor and control your security system right from your smartphone. This means you'll receive real-time notifications and have the peace of mind that your home is always protected, even when you're away.

2. Components List

- ESP32 Development Board
- PIR Sensor
- Buzzer
- LEDs
- Breadboard
- Jumper Wires
- USB Cable
- Power Supply
- Blynk Application

3. Component Description

ESP32 Development Board: This powerful microcontroller comes with built-in Wi-Fi and Bluetooth capabilities, making it ideal for IoT projects. It features multiple GPIO pins, analog inputs, and supports various communication protocols like I2C, SPI, and UART. The ESP32 can be programmed using the Arduino IDE, and its Wi-Fi feature allows for easy integration with cloud services and remote monitoring applications.

PIR Sensor: The Passive Infrared (PIR) sensor detects motion by measuring the infrared radiation emitted by objects in its field of view. When an object, such as a human or animal, moves within the sensor's range, it triggers a change in the infrared radiation levels, causing the sensor to activate. PIR sensors are commonly used in security systems, automatic lighting, and occupancy detection.

 Buzzer: This small electronic component produces a loud sound when activated, serving as an audible alert. Buzzers come in two types: active and passive. An active buzzer generates sound when power is applied, while a passive buzzer requires an external signal to produce sound. In your project, the buzzer will sound an alarm when motion is detected by the PIR sensor.

 LEDs: Light Emitting Diodes (LEDs) are semiconductor devices that emit light when an electric current passes through them. They are available in various colors, and in your project, they will provide visual indicators. For example, a red LED can indicate an alert when motion is detected, while a green LED can show the system's operational status. LEDs are energy-efficient and have a long lifespan.

 Breadboard: This reusable prototyping board features a grid of interconnected holes and strips, allowing you to easily connect electronic components without soldering. The breadboard provides a convenient platform for building and testing your circuits before making permanent connections. It consists of two main areas: a terminal strip for component connections and a bus strip for power distribution.

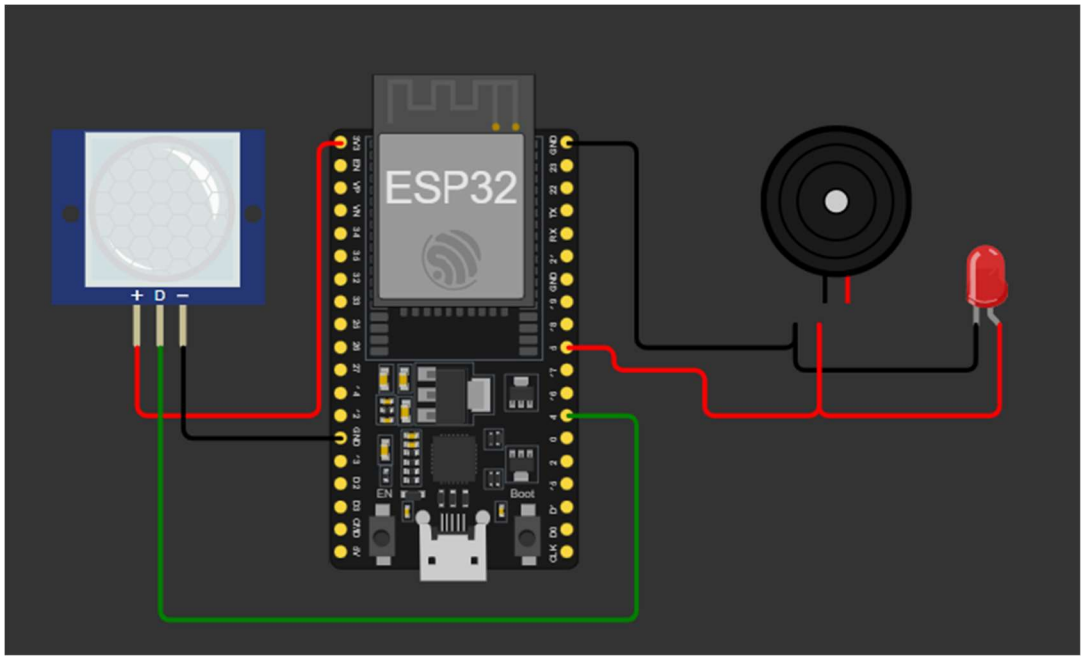
 Jumper Wires: These flexible wires are used to make connections between components on the breadboard and the ESP32. Jumper wires come in various lengths and colors, with male or female connectors on the ends. They are essential for quickly and easily setting up your prototype circuit.

 USB Cable: This cable is used to connect the ESP32 to your computer for programming and power. The most common type is a USB-A to Micro-USB cable. The USB connection allows you to upload code to the ESP32 using the Arduino IDE and provides power to the board during development.

 Power Supply: If you prefer not to use the USB connection for powering the ESP32, you can use an external power supply. The ESP32 typically operates at 3.3V, but many development boards come with built-in voltage regulators to accept a wider range of input voltages (e.g., 5V or 9V). Ensure that your power supply meets the voltage and current requirements of the ESP32.

 Blynk Application: Blynk is a mobile app that enables remote monitoring and control of IoT devices. By integrating the Blynk app with your ESP32, you can create a user-friendly interface on your smartphone to receive real-time notifications, monitor sensor data, and control the system remotely. Blynk supports both Android and iOS devices and provides a wide range of widgets for building custom dashboards.

4. Circuit Diagram



5. Working

The Intrusion Detection System (IDS) project is all about keeping your home safe and secure. By using an ESP32 microcontroller paired with a PIR sensor, this system can detect motion, sound an alarm with a buzzer, and provide visual alerts with LEDs. When the system is powered on, it connects to your Wi-Fi network and links to the Blynk app using a unique authentication token.

Whenever the PIR sensor detects any movement, the ESP32 springs into action, activating the buzzer to sound an alert and updating the Blynk app to send real-time notifications straight to your smartphone. The LEDs also light up to indicate the system's status, ensuring you're always in the loop. The PIR sensor continuously monitors for motion, and the buzzer stops once the motion ceases.

With this project, you can enhance your home security by remotely monitoring and controlling the system through the Blynk app. It provides peace of mind by promptly alerting you of any unauthorized entry, so you can feel confident that your home is protected.

6. Software code

```
#define BLYNK_TEMPLATE_ID "TMPL30r3WgH8A"

#define BLYNK_TEMPLATE_NAME "theft alert"
```

```
#define BLYNK_AUTH_TOKEN "67iHJubkY2qo5cO36sbvXsWJNuZgr4Lj"

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Your WiFi credentials.
char ssid[] = "Wifi SSID";
char pass[] = "WIFI Password";

const int PIR_SENSOR_PIN = 4;    // Pin connected to the PIR sensor OUT pin
const int BUZZER_PIN = 5;        // Pin connected to the buzzer positive (+) pin
int pinStateCurrent   = LOW;     // current state of pin
int pinStatePrevious  = LOW;     // previous state of pin

BlynkTimer timer;

// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED() {
  Blynk.syncVirtual(V0);  // will cause BLYNK_WRITE(V0) to be executed
```

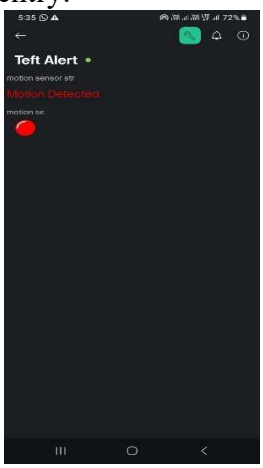
```
  Blynk.syncVirtual(V1);  // will cause BLYNK_WRITE(V1) to be executed
}

void setup() {
  // Debug console
  Serial.begin(115200);  // initialize serial
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  pinMode(PIR_SENSOR_PIN, INPUT);  // set ESP32 pin to input mode to read value from OUTPUT pin of sensor
  pinMode(BUZZER_PIN, OUTPUT);     // Initialise digital pin for buzzer as an output pin

  // Give 3 little buzzer sounds to indicate the hardware is ready to use
  for (int i = 0; i < 3; i++) {
    digitalWrite(BUZZER_PIN, HIGH);  // Turn on the buzzer
    delay(100);  // Delay for 100 milliseconds
    digitalWrite(BUZZER_PIN, LOW);   // Turn off the buzzer
    delay(100);  // Delay for 100 milliseconds
  }
}

void loop() {
  Blynk.run();
```

```
  timer.run();


  pinStatePrevious = pinStateCurrent; // store old state

  pinStateCurrent = digitalRead(PIR_SENSOR_PIN);   // read new state


  Serial.print("Previous state: ");

  Serial.println(pinStatePrevious);

  Serial.print("Current state: ");

  Serial.println(pinStateCurrent);


  if (pinStatePrevious == LOW && pinStateCurrent == HIGH) {   // pin state change:
LOW -> HIGH

    Serial.println("Motion detected!");

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

    Blynk.virtualWrite(V0, 1);

    Blynk.virtualWrite(V1, "Motion Detected");

  } else if (pinStatePrevious == HIGH && pinStateCurrent == LOW) {   // pin state
change: HIGH -> LOW

    Serial.println("Motion stopped!");

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

    Blynk.virtualWrite(V0, 0);

    Blynk.virtualWrite(V1, "Motion Stopped");

  }

}
```

7. Results

The Intrusion Detection System (IDS) project results in a smart security system that detects motion using a PIR sensor. When motion is detected, the ESP32 triggers a buzzer to sound an alarm, sends real-time notifications to the user's smartphone via the Blynk app, and uses LEDs for visual alerts. This system enhances home security by providing remote monitoring and immediate alerts of any unauthorized entry.

When Motion is not detected



When Motion is detected

8. Applications

- **Home Security**: Detects unauthorized entry and sends real-time alerts to homeowners,ensuring immediate response and peace of mind.

- **Office and Commercial Security**: Monitors office buildings, shops, and commercial spaces to prevent theft and unauthorized access, protecting valuable assets.

- **Warehouse and Storage Monitoring**: Secures warehouses and storage facilities by detecting any movement, reducing the risk of theft and vandalism.

- **Elderly and Vulnerable Person Monitoring**: Monitors the movement of elderly or vulnerable individuals, alerting caregivers in case of unusual activity or emergencies.

- **Pet Monitoring**: Tracks the movement of pets within the home, providing alerts if they enter restricted areas.

9. References

https://youtu.be/8Umvp2rbcJg?si=Ic-NwOSKp3jmUJHL

https://wokwi.com/projects/new/esp32

https://copilot.microsoft.com/