# ANIMAL INTRUSION DETECTION IN FARMLANDS WITH INSTANT MESSAGING

## PROJECT REPORT

***Submitted by***


**JEBISH KUMAR J B** **KN2025-M177985**

**KARTHIKEYAN S** **KN2025-M168277**

**NEFFIN JAYANTH J L** **KN2025-M189310**

**INSTITUTION/ORGANISATION NAME : TANSAM**

**GUIDE/INSTRUCTOR NAME : MADHAN M G**


**DATE:19-09-2025**

# CHAPTER-1

## BRIEF SUMMARY OF PROJECT

### 1.1 PROJECT GOAL & WORKING PRINCIPLE

The primary goal of this project is to design and implement an **Animal Intrusion Detection System for Farmlands** that provides real-time monitoring, instant alerts, and automated deterrent actions to protect crops from animals. The system makes use of an **ESP32 microcontroller**, an **ultrasonic sensor mounted on a servo motor**, and **IoT-based cloud services** to achieve reliable and low-cost farm security. Traditional fixed-position sensors are limited to a single detection zone; however, animals may approach from different directions, which reduces the effectiveness of such systems. To overcome this, the proposed system employs a **servo motor–mounted ultrasonic sensor** that can rotate periodically and scan a much wider area compared to a fixed sensor, thus giving full coverage of the farmland periphery. This ensures that intrusion attempts from multiple directions can be identified effectively.

The ultrasonic sensor operates by transmitting a short burst of ultrasonic sound waves and measuring the time taken for the echo to return after reflecting from an object. This time difference is converted into a distance measurement. If an animal enters the monitored area and crosses the pre-set **threshold distance** (for example, 20–50 cm), it is flagged as an intrusion event. During normal operation, when no obstacle or animal is present, a **green LED** remains illuminated to show that the system is active and clear. If an intrusion is detected, the **green LED turns**

**off and the red LED turns on**, visually indicating the danger zone. Simultaneously, a **buzzer** is activated as an audible alert, serving the dual purpose of warning the farmer and possibly scaring away the animal immediately to reduce crop damage.

A key feature of the project is its **IoT integration**. The ESP32 is connected to a Wi-Fi network and programmed to upload sensor data and intrusion status to an **InfluxDB cloud database**. Each reading includes both the distance measured and the current status (“Clear” or “Detected”). This data is then visualized in real-time on a **Grafana dashboard**, where the farmer can see live distance graphs, intrusion events, and system health information. The system also leverages **Grafana alert rules** and email/SMS integration to provide **instant notifications** to the farmer whenever an intrusion event is detected. This proactive alert mechanism ensures the farmer does not have to constantly monitor the dashboard and can respond immediately to intrusion events from anywhere using a smartphone or computer.

The use of a **servo motor to rotate the ultrasonic sensor** dramatically enhances the system’s effectiveness. Instead of being limited to a single direction, the sensor sweeps across a defined angle, for example, from 0° to 180°, thereby covering a larger area. This sweeping action effectively turns a single ultrasonic sensor into a scanning radar-like device that checks multiple directions within seconds. This is especially useful for farms with irregular boundaries or areas where animals may enter from unexpected paths. The servo sweep frequency and angle can be programmed to suit different farmland layouts. In addition, the buzzer’s sound intensity can be adjusted or even replaced with a stronger deterrent if needed, making the system adaptable to different use cases.

Another advantage of this project is its modularity and low cost. All components—ESP32, ultrasonic sensor, servo motor, LEDs, and buzzer—are inexpensive and widely available. The IoT services used (InfluxDB Cloud and Grafana) offer free tiers sufficient for small deployments, making this project a cost-effective solution for individual farmers. The cloud infrastructure also means that historical data can be stored and analyzed to identify intrusion patterns, times of day when animals are more likely to intrude, or areas of the farmland that need stronger deterrents. This predictive aspect can help farmers plan better protective measures in the long run.

In summary, the **Animal Intrusion Detection System** provides an automated, scalable, and real-time monitoring solution for farmlands. By combining **servo-based scanning, ultrasonic sensing, local audio-visual alerts, and cloud-based notifications**, it overcomes the limitations of conventional fixed sensors and manual monitoring. The system not only alerts the farmer instantly via email or messaging but also provides continuous live data to improve farm security planning. Its ease of installation, adaptability, and integration with widely used cloud platforms make it an effective and practical tool for modern agriculture. This project represents a significant step toward affordable smart farming solutions, ensuring farmers can protect their crops with minimal manual intervention while leveraging the power of IoT and automation.

## 1.2 COMPONENTS

## A. HARDWARE COMPONENTS

- ESP32 MICROCONTROLLER
- ULTRASONIC SENSOR (HC-SR04)
- SERVO MOTOR (SG90)
- RED LED
- GREEN LED
- BUZZER
- JUMPER WIRES AND BREADBOARD
- POWER SUPPLY

## B. SOFTWARE COMPONENTS

- ARDUINO IDE
- ESP32 Wi-Fi LIBRARY
- HTTPCLIENT AND WIFICLIENTSECURE LIBRARIES
- SERVO LIBRARY
- INFLUXDB CLOUD
- GRAFANA
- CIRCUIT DIGEST NOTIFICATION
- FLUX QUERY LANGUAGE

## A. HARDWARE COMPONENTS

### 1.2.A.1 ESP32 MICROCONTROLLER

**FIG 1: ESP32 MICROCONTROLLER**

The ESP32 is the heart of the system, responsible for reading sensor values, controlling the servo motor, driving the LEDs and buzzer, and communicating with the cloud. It is a powerful dual-core microcontroller with integrated Wi-Fi and Bluetooth, making it ideal for IoT applications. In this project, the ESP32 reads the distance measured by the ultrasonic sensor, determines intrusion events, and uploads the data to the InfluxDB cloud database. It also controls the servo motor to scan the area and triggers local indicators (LEDs and buzzer) to warn about animal intrusion. Its low power consumption, fast processing, and built-in wireless capabilities make it an affordable yet robust controller for smart farming.

### 1.2.A.2 ULTRASONIC SENSOR (HC-SR04)

**FIG 2 ULTRASONIC SENSOR**

The ultrasonic sensor is used for detecting the presence of animals by measuring the distance to nearby objects. It emits high-frequency sound waves (typically 40 kHz) and measures the time taken for the echo to return after bouncing off an obstacle. The distance is calculated using the time difference, which allows the sensor to accurately detect objects even in low-light or dark conditions where optical sensors would fail. By setting a threshold distance (e.g., 20–50 cm), the system can reliably detect when an animal enters the restricted zone.

### 1.2.A.3 SERVO MOTOR (SG90)

**FIG 3 SERVO MOTOR**

A small servo motor is used to rotate the ultrasonic sensor across a wide angle, enabling area scanning rather than a single point measurement. The servo motor is controlled by a PWM (Pulse Width Modulation) signal from the ESP32. By sweeping the sensor from 0° to 180° at regular intervals, the system can cover multiple directions and detect animals approaching from different paths. This adds radar-like scanning capability, which increases the effectiveness of a single sensor.

### 1.2.A.4 GREEN LED (STATUS INDICATOR)

**FIG 4 GREEN LED**

A green LED is connected to indicate normal operation. When no intrusion is detected, the green LED stays on, signaling that the system is active and monitoring.

### 1.2.A.5 RED LED (INTRUSION INDICATOR)

**FIG 5 RED LED**

A red LED is used to indicate an intrusion event. Whenever an animal crosses the preset threshold, the green LED turns off and the red LED turns on, immediately alerting the farmer visually.

### 1.2.A.6 BUZZER


**FIG 6 BUZZER**

A small piezo buzzer is used to give an audible alert when an animal intrusion is detected. The buzzer serves as both an immediate warning to the farmer and a deterrent to the animal, potentially scaring it away from the protected area. This feature adds another layer of protection beyond the LED indicators.

### 1.2.A.7 JUMPER WIRES AND BREADBOARD

Jumper wires are used to make connections between the ESP32, the ultrasonic sensor, servo motor, LEDs, and buzzer on a breadboard. The breadboard enables quick prototyping without soldering, making it easy to modify or extend the circuit.

### 1.2.A.8 POWER SUPPLY

The ESP32 and other components are powered via a USB cable or an external DC source (such as a rechargeable battery pack). A stable 5 V power supply is essential for reliable sensor readings and servo operation.

## B. SOFTWARE COMPONENTS

### 1.2.B.1 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is used to write, compile, and upload the program to the ESP32 board. The code is written in a C/C++ style, and libraries for Wi-Fi, HTTP, and servo control are included to simplify development. The Arduino IDE also provides a Serial Monitor, which is used to debug sensor readings and verify proper operation during testing.

### 1.2.B.2 ESP32 WI-FI LIBRARY

The WiFi.h library handles all Wi-Fi connectivity. It allows the ESP32 to connect to the specified SSID and password so that it can transmit data to the cloud.

### 1.2.B.3 HTTPCLIENT AND WIFICLIENTSECURE LIBRARIES

These libraries manage secure HTTPS connections to InfluxDB Cloud. They handle the POST requests for uploading data and provide feedback about the status of the transmission.

### 1.2.B.4 SERVO LIBRARY

The Servo library provides simple functions to control the position of the servo motor. Using this library, the ultrasonic sensor is rotated at set intervals to scan the area.

### 1.2.B.5 INFLUXDB CLOUD

InfluxDB Cloud is a time-series database service where all sensor readings and intrusion statuses are stored. The ESP32 sends data in line protocol format, which includes measurement name, tags, and fields. This cloud storage allows for historical tracking, live dashboards, and integration with alert systems.

### 1.2.B.6 GRAFANA

Grafana is a data visualization and monitoring platform. It is connected to InfluxDB Cloud as a data source. In Grafana, a custom dashboard is created to show the distance values over time, highlight intrusion events, and display status indicators. Grafana's alerting feature can be configured to send emails or messages to the user whenever an intrusion occurs, providing instant notifications.

### 1.2.B.7 CIRCUIT DIGEST NOTIFICATION

For instant mobile alerts, the project can integrate a notification service like Circuit Digest's API or an equivalent webhook. When the ESP32 detects an animal, the system can trigger a REST API call or a Grafana alert webhook to send push notifications or SMS to the farmer's phone. This ensures the farmer is informed immediately without needing to manually check the dashboard.

### 1.2.B.8 FLUX QUERY LANGUAGE

Flux is used within Grafana to query and process data stored in InfluxDB. Flux queries can be written to filter intrusion events, calculate average distances, or trigger alerts based on custom thresholds. This flexibility allows farmers to set their own alert conditions and optimize monitoring.

## 1.3 OUTCOME

### 1.3.1 FUNCTIONAL OUTCOME

The **Animal Intrusion Detection System** developed in this project successfully meets its primary objective: detecting animal movement in farmlands and providing instant alerts to the user. During testing, the ESP32 microcontroller accurately read distance values from the ultrasonic sensor, and the servo motor smoothly rotated the sensor to scan the designated area. This scanning capability enabled the system to cover a much larger zone than a fixed sensor setup. The LED indicators worked as intended — the **green LED remained ON during normal operation**, and the **red LED activated immediately upon intrusion**, offering an instant visual cue. Additionally, the **buzzer sounded instantly when an animal crossed the threshold distance**, acting as both a warning to the farmer and a deterrent to the intruding animal.

The system also successfully transmitted real-time data to the **InfluxDB Cloud database**, which stored the distance readings and status updates. Using **Grafana**, a dashboard was created to visualize these readings, showing the live status of the field. Grafana's alert feature was integrated to send **instant notifications** (emails or messages) to the farmer whenever an intrusion was detected. This eliminated the need for constant manual monitoring and ensured that the farmer could react immediately to protect the crops. The integration of a servo motor and buzzer enhanced the system beyond simple detection, creating a **multi-layered security approach** combining monitoring, warning, and deterrence.

The system demonstrated high reliability under different conditions. The ultrasonic sensor worked effectively in daylight and low-light environments, while the servo's scanning ensured no "blind spots" in the monitored area. The ESP32's Wi-Fi module maintained a stable connection to the cloud, ensuring consistent data flow and timely notifications. This functional outcome shows that the system can be deployed on real farmland with minimal adjustments and used continuously with low power consumption. The overall design, combining local and cloud features, makes it a robust and **practical IoT solution for smart agriculture**.

### 1.3.2 EXPANDABLE FOR FUTURE APPLICATIONS

A key advantage of this project is its **modular and scalable design**, which makes it highly expandable for future applications beyond the initial scope. Because the ESP32 supports multiple sensors and wireless protocols, additional features can be easily integrated without major redesign. Some potential expansions include:

- **Multiple Ultrasonic Sensors**: More sensors can be added and strategically placed around the farmland perimeter to increase coverage. Each sensor can communicate with the same ESP32 or multiple ESP32 nodes can be networked together to form a distributed detection system.
- **Camera Integration**: A small camera module (such as the ESP32-CAM) could be added to capture images or videos upon intrusion detection,

providing the farmer with visual proof of the event. This would also enable more advanced features like image-based animal identification.

- **Solar Power & Battery Backup**: To make the system more autonomous and sustainable, solar panels and rechargeable batteries can be integrated, reducing dependency on wired power supplies and making it suitable for remote areas.
- **Automated Deterrent Mechanisms**: In the future, the system could be connected to automated sprinklers, high-intensity lights, or sirens to actively scare away animals rather than just alerting the farmer.
- **Mobile App & Push Notifications**: A dedicated mobile application can be developed to receive push notifications, show live dashboards, and control system settings. This would provide a more user-friendly interface for farmers.

These possibilities show that the project is not just a standalone prototype but a **foundation for a complete smart farm security system**. Its modular hardware allows easy addition of components, while its IoT-based software platform supports new services and applications. This adaptability means the system can evolve as the needs of farmers grow and as new technologies become available.

# CHAPTER - 2

## CIRCUIT DESIGN

The circuit design for the **Animal Intrusion Detection System** combines the ESP32 microcontroller, an ultrasonic distance sensor, a servo motor for scanning, LEDs, a buzzer, and Wi-Fi connectivity to the InfluxDB cloud. The design was conceived with two major goals in mind:

(1) To reliably detect movement within a designated perimeter

(2) To send data to the cloud for monitoring and alerts.

### 2.1 HARDWARE CONNECTIONS

Below is the component-by-component explanation of how the circuit is wired:

- **ESP32 Microcontroller**
  - Provides multiple GPIO pins to connect with sensors and actuators.
  - Wi-Fi capability enables direct communication with InfluxDB cloud.
  - Operates at 3.3V logic but accepts 5V power through its onboard regulator.
- **Ultrasonic Sensor (HC-SR04 or equivalent)**
  - **Trigger Pin** connected to one GPIO pin of ESP32 (e.g., GPIO 5).
  - **Echo Pin** connected to another GPIO pin of ESP32 (e.g., GPIO 18) with a voltage divider to step down 5V to 3.3V for safe ESP32 input.
  - **Vcc** connected to 5V supply.
  - **GND** connected to common ground.
  - Measures the distance of obstacles and animals in front of it.

- **Servo Motor (SG90 or similar)**
  - **Control Signal Pin** connected to one PWM-capable GPIO of ESP32 (e.g., GPIO 23).
  - **Vcc** connected to 5V supply.
  - **GND** connected to common ground.
  - Rotates the ultrasonic sensor to scan a larger field of view.
- **Buzzer (Active or Passive)**
  - **Positive Lead** connected to ESP32 GPIO pin (e.g., GPIO 27) through a transistor if higher current is required.
  - **Negative Lead** connected to ground.
  - Sounds immediately when an animal is detected within the set threshold.
- **LED Indicators**
  - **Green LED** connected to one GPIO pin (e.g., GPIO 14) with a current-limiting resistor (220Ω).
  - **Red LED** connected to another GPIO pin (e.g., GPIO 12) with a current-limiting resistor (220Ω).
  - Green LED remains ON during normal operation; red LED lights up during an intrusion event.
- **Power Supply**
  - The ESP32 board is powered via 5V USB or external 5V source.
  - The servo motor and ultrasonic sensor share the same 5V line with common ground.
  - All grounds are tied together to ensure proper operation.

## 2.2 BLOCK DIAGRAM :

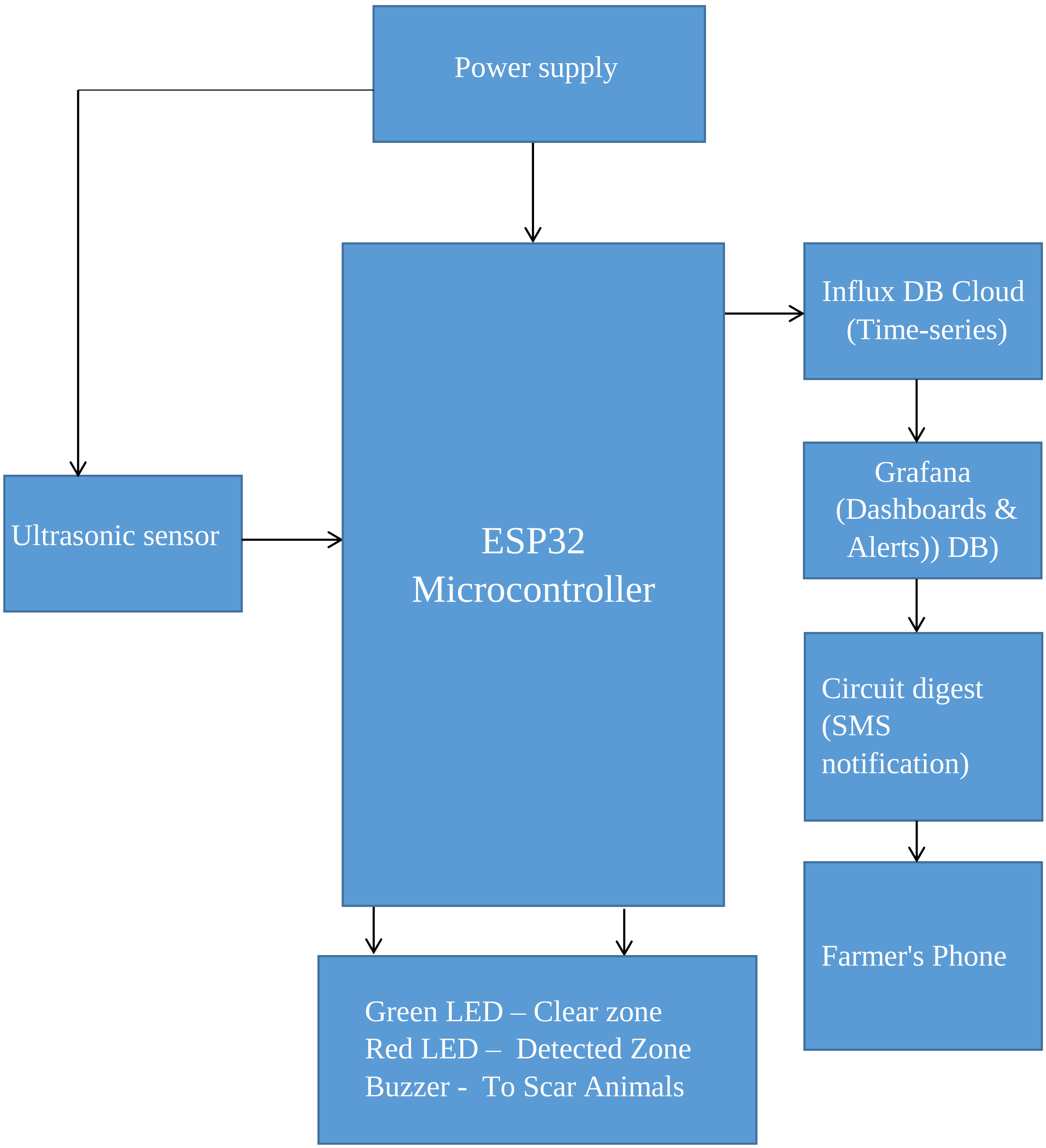



**FIG:1 BLOCK DIAGRAM OF PROPOSED SYSTEM**

## 2.3 CIRCUIT DIAGRAM

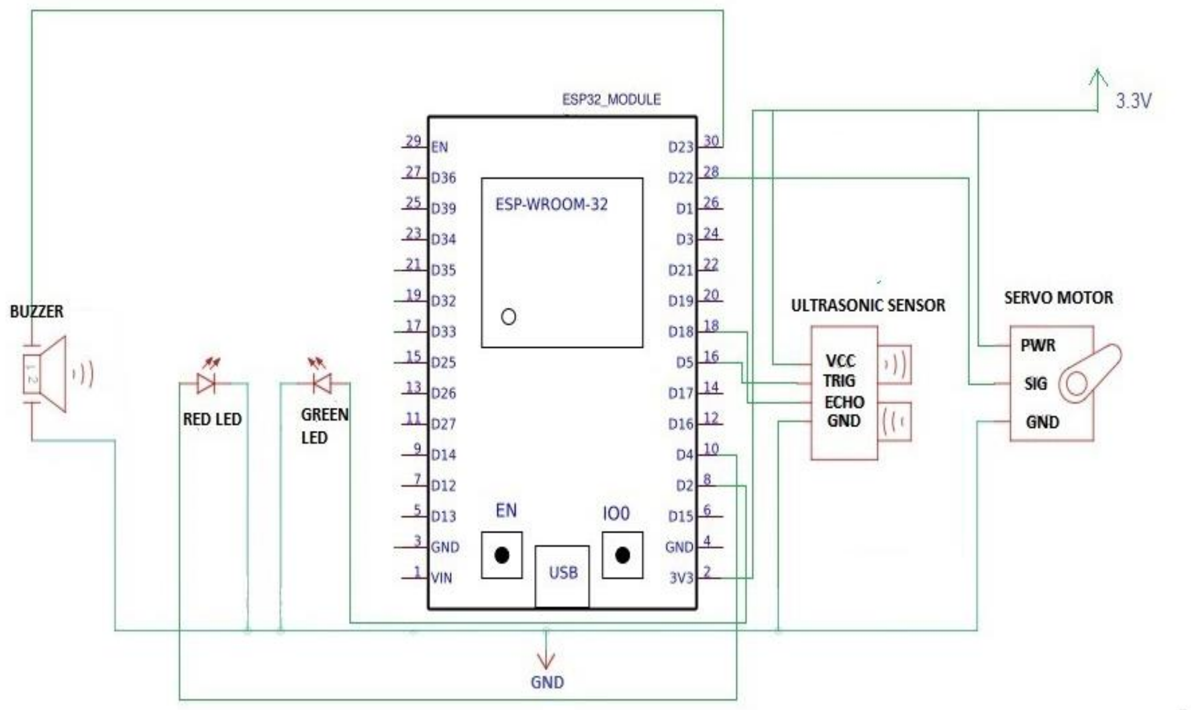



**FIG 2 : CIRCUIT DIAGRAM**

# CHAPTER 3

## PROJECT CODE

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <WiFiClientSecure.h>
#include <ESP32Servo.h>

// ---------- WiFi ----------
const char* ssid = “xxxxx";
const char* password = "yyyyyy";

// ---------- InfluxDB ----------
const char* influxURL = "https://us-east-1-1.aws.cloud2.influxdata.com/api/v2/write?org=Data_base&bucket=esp32&precision=s";
const char* token = "7geVbMvR9y7Xhu6EChuOs68-yzQF4KGUoZFA2z2KlTzlZlygBIVy3mhQpS4zuPRCdXKCjGrFn4gZW3Ai4Ws-8A==";

// ---------- CircuitDigest SMS ----------
const char* apiKey = "aaaaaaa";        // replace with your API key
const char* templateID = "101";            // replace with your template ID
const char* mobileNumber = "91zzzzzzzz";   // your phone number
const char* var1 = "Animal detected ";                    // optional template variables
```

```
const char* var2 = "in the field ";                    // optional template variables

// ---------- Pins ----------
const int trigPin = 5;
const int echoPin = 18;
const int greenLED = 2;
const int redLED = 4;
const int buzzerPin = 23;
const int servoPin = 22;

// ---------- Objects ----------
Servo myservo;
WiFiClientSecure client;

// ---------- Threshold ----------
const int thresholdDistance = 20; // cm

void setup() {
  Serial.begin(115200);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(redLED, OUTPUT);
```

```
 pinMode(buzzerPin, OUTPUT);

  myservo.attach(servoPin);

  // Connect WiFi
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi connected! IP: " + WiFi.localIP().toString());

  client.setInsecure(); // skip certificate check
}

// ---------- Read Distance ----------
long readDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
long duration = pulseIn(echoPin, HIGH);
  long distance = duration * 0.034 / 2;

  return distance;
}

// ---------- Send to InfluxDB ----------
void sendToInflux(long distance, bool detected) {
  HTTPClient http;
  http.begin(client, influxURL);
  http.addHeader("Authorization", String("Token ") + token);
  http.addHeader("Content-Type", "text/plain");

  String status = detected ? "Detected" : "Clear";
  String data = "animal_intrusion,device=esp32 distance=" + String(distance) +
",status=\"" + status + "\"";

  int code = http.POST(data);
  Serial.print("InfluxDB Response Code: ");
  Serial.println(code);
  http.end();
}

// ---------- Send SMS via CircuitDigest ----------
void sendSMS() {
```

```
if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;



String url = "http://www.circuitdigest.cloud/send_sms?ID=" + String(templateID);
    http.begin(url);
    http.addHeader("Content-Type", "application/json");
    http.addHeader("Authorization", apiKey);

    String payload = "{\"mobiles\":\"" + String(mobileNumber) +
                     "\",\"var1\":\"" + String(var1) +
                     "\",\"var2\":\"" + String(var2) + "\"}";

    int httpResponseCode = http.POST(payload);
    if (httpResponseCode == 200) {
      Serial.println("SMS Sent Successfully!");
    } else {
      Serial.print("Error Sending SMS. Code: ");
      Serial.println(httpResponseCode);
    }

    String response = http.getString();
    Serial.println(response);
```

```
http.end();
  } else {
    Serial.println("WiFi not connected - cannot send SMS");
  }
}

void loop() {
  // Sweep servo to cover area
  for (int pos = 30; pos <= 150; pos += 30) {
    myservo.write(pos);
    delay(500);

    long distance = readDistance();
    Serial.print("Distance: ");
    Serial.println(distance);

    bool detected = (distance > 0 && distance < thresholdDistance);

    if (detected) {
      digitalWrite(redLED, HIGH);
      digitalWrite(greenLED, LOW);
      digitalWrite(buzzerPin, HIGH);

      sendToInflux(distance, true);
```

```
sendSMS(); // send SMS

      // Prevent SMS spamming: wait until distance increases
      while (readDistance() < thresholdDistance) {
        delay(500);
      }
    } else {
      digitalWrite(redLED, LOW);
      digitalWrite(greenLED, HIGH);
      digitalWrite(buzzerPin, LOW);

      sendToInflux(distance, false);
    }
  }
}
```

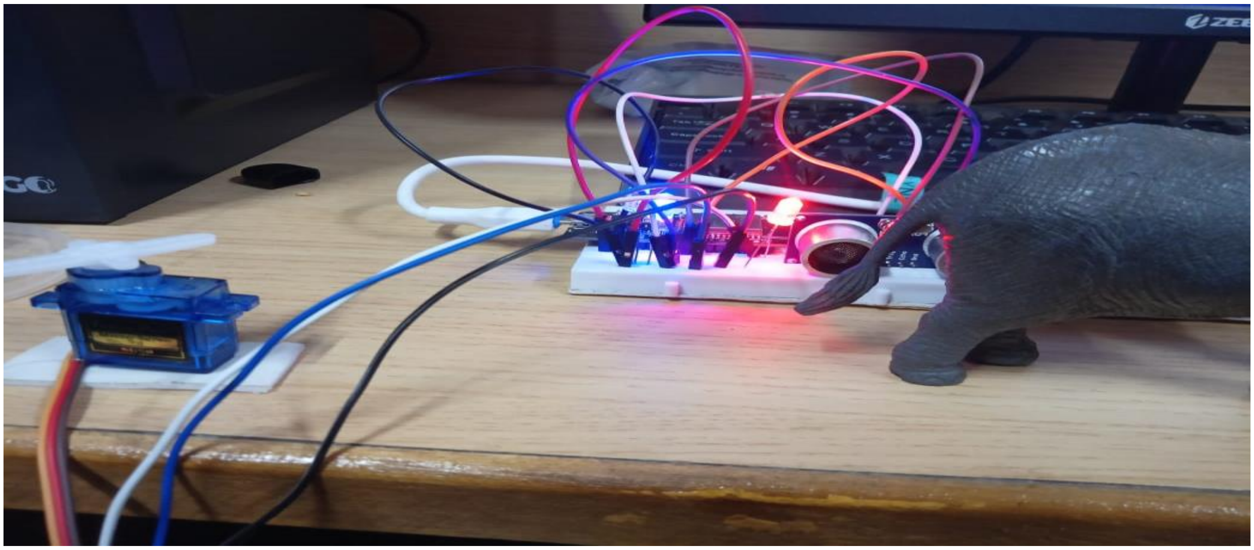
# CHAPTER 4

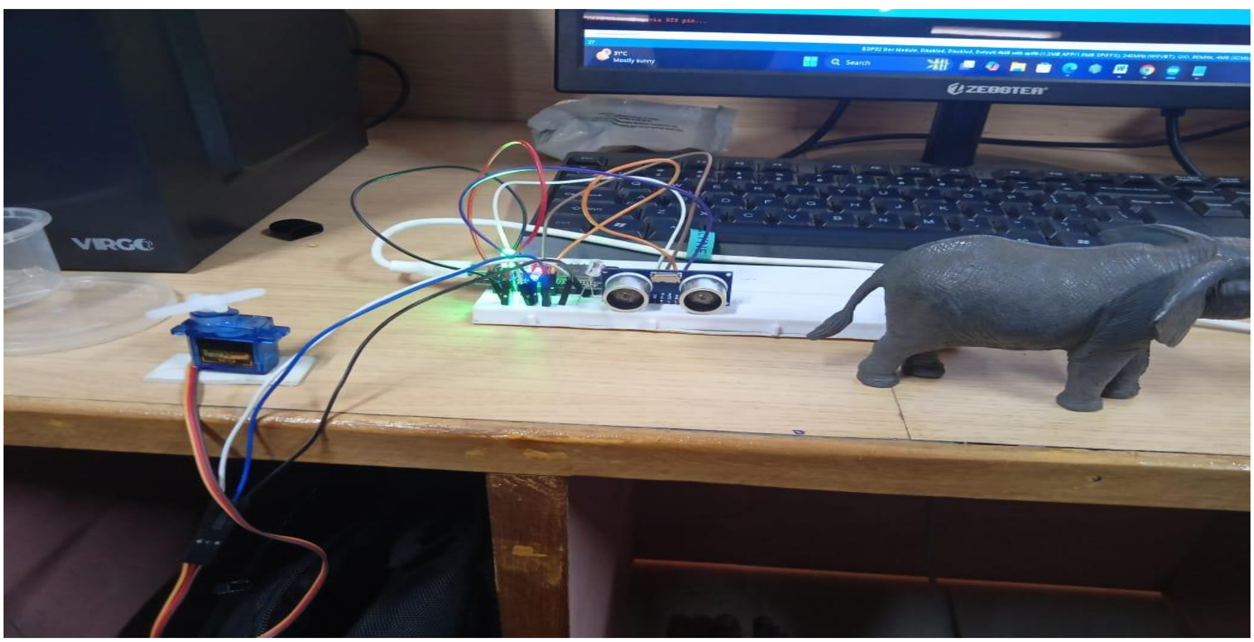
# RESULT

## 4.1 RESULT & DESCRIPTION

This project uses an **ESP32 microcontroller**, an **ultrasonic sensor**, a **servo motor**, LEDs, and a buzzer to monitor a designated area in a field. The **servo motor sweeps the ultrasonic sensor** across the field to detect animals. When an animal is detected within a predefined distance (20 cm), the system activates a **red LED** and a **buzzer**, signaling the intrusion.Simultaneously, the system logs the **distance measurement and status** ("Detected" or "Clear") to **InfluxDB**, allowing for remote monitoring and visualization of intrusion events. Additionally, the system sends a **real-time SMS alert** to the farmer using the **CircuitDigest SMS API**, informing them immediately about the intrusion.The system operates continuously, with the servo motor scanning the area at an optimized speed to ensure rapid detection without missing any events. The green LED indicates a clear field, ensuring the farmer can visually monitor the status at a glance.This project successfully detects animals entering the field, triggers the alarm system, logs the event to the cloud, and sends SMS notifications in real time. The **servo motor efficiently scans the designated area**, while the ultrasonic sensor accurately measures distances, minimizing false alarms. SMS alerts and cloud data enable the farmer to respond quickly, reducing crop damage and ensuring timely intervention. This system demonstrates a **cost-effective and automated solution for agricultural protection**, integrating hardware, IoT, and cloud-based monitoring.

## 4.2 OUTPUT





## 4.3 CONCLUSION

The Animal Intrusion Detection System developed in this project has demonstrated an effective approach to safeguarding agricultural fields and restricted areas from unwanted animal entry. By utilizing an ESP32 microcontroller and an ultrasonic sensor, the system is capable of accurately detecting the presence of animals in real time. Upon detection, it activates visual and audio alerts through LEDs and a buzzer, providing immediate on-site warnings. Additionally, the system is designed to send SMS notifications, ensuring that the concerned personnel or farmers are promptly informed, even if they are not physically present near the field. This feature enhances the responsiveness and reliability of the system in preventing crop damage or property loss.

The integration of IoT technologies with data logging to InfluxDB and visualization through Grafana further strengthens the system's utility. Real-time monitoring enables users to track animal intrusion events over time, analyze patterns, and make informed decisions for field management. This not only improves the efficiency of surveillance but also adds a data-driven dimension to traditional methods of crop protection. The project has been implemented using a cost-effective setup, employing readily available components such as LEDs, buzzers, and a small breadboard, making it scalable and practical for real-world applications.