

PANDIT DEENDAYAL ENERGY UNIVERSITY
SCHOOL OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INTERNET OF THINGS LAB
COURSE PROJECT FILE



PROJECT TITLE: IoT-Based Home Automation system

NAME: Vraj Patel, Harsh shah

ROLL NO: 21BCP362, 21BCP359

SEMESTER: 7

INDEX

Sr. No.	Title	Page
1.	Introduction	3
2.	Methodology	4
3.	Circuit and code	6
4.	Results and Conclusion	10

Chapter 1: Introduction

Overview

This project is a comprehensive Home Automation System based on the ESP32 microcontroller, utilizing IoT capabilities to monitor and control various sensors and actuators remotely via the Blynk platform. The setup includes an LDR (Light Dependent Resistor) for light sensitivity detection, a water sensor for water level monitoring, an LED for status indication, and a buzzer for audible alerts. The entire system is controlled and monitored in real-time through the Blynk mobile app, offering a user-friendly interface for remote access.

Objective

The primary objective of this project is to create a cost-effective, scalable, and efficient home automation system that enhances safety and convenience. The system aims to:

- Automatically control the LED based on ambient light levels.
- Provide real-time monitoring of water levels to prevent overflow.
- Enable remote control of the LED and buzzer through the Blynk app.
- Alert users in case of low light or high water level conditions.

Technology Used

- ESP32 Microcontroller: A powerful, WiFi-enabled microcontroller that serves as the core of the project, handling all sensor readings and Blynk communication.
- Blynk Platform: An IoT platform used for creating the mobile app interface, enabling real-time remote control and monitoring.
- LDR Sensor: Used for detecting ambient light levels and automatically controlling the LED.
- Water Sensor (HW-038): Monitors water levels and triggers alerts if the level exceeds a threshold.
- LED and Buzzer: Actuators for visual and audible alerts, controlled both automatically by sensor input and manually through the Blynk app.

This project demonstrates the potential of IoT in creating smart home solutions, focusing on safety, convenience, and energy efficiency.

Chapter 2: Methodology

1. System Design

The system is designed around the ESP32 microcontroller, integrating both sensors and actuators. It features:

- **LDR Sensor:** For light sensitivity detection, automatically controlling the LED based on ambient light.
- **Water Sensor:** For monitoring water levels and triggering alerts if the level is too high.
- **LED and Buzzer:** Provide visual and audible feedback, controlled manually via Blynk or automatically by sensor input.
- **Blynk App:** Offers a user-friendly interface for remote control and real-time monitoring.

2. Hardware

- **ESP32 Microcontroller:** The main controller with built-in WiFi for IoT capabilities.
- **LDR (Light Dependent Resistor):** Senses ambient light and outputs an analog value.
- **Water Sensor (HW-038):** Detects water levels and outputs an analog value.
- **LED:** Indicates system status and provides alerts.
- **Buzzer:** Provides audible alerts for high water levels.
- **Resistors:** Used in the LDR voltage divider circuit (220-ohm resistor).

3. Software

- **Arduino IDE:** Used for programming the ESP32 microcontroller.
- **Blynk Platform:** Provides the mobile app interface for remote control and monitoring.
- **Libraries:** Includes WiFi.h for network connection and BlynkSimpleEsp32.h for Blynk integration.

4. Implementation Steps

Hardware Setup:

- Connect the LDR with a 220-ohm resistor as a voltage divider.
- Connect the water sensor, LED, and buzzer to the designated GPIO pins.
- Power the ESP32 and connect the components as per the circuit diagram.

Software Development:

- Write the code in the Arduino IDE, including sensor reading functions and Blynk integration.
- Upload the code to the ESP32 using the Arduino IDE.

Blynk App Configuration:

- Create a new Blynk project, set up virtual pins, and configure buttons for LED and buzzer control.
- Link the app with the ESP32 using the Blynk Auth Token.

Testing and Calibration:

- Test each component individually (LDR, water sensor, LED, and buzzer).
- Calibrate the LDR sensor for accurate light sensitivity readings and adjust thresholds as needed.

Workflow

- The system initializes and connects to WiFi using the Blynk Auth Token.
- The LDR sensor reads ambient light levels, scales the value, and sends it to the Blynk app. If light levels are low, the LED turns on automatically.
- The water sensor checks the water level and sends the value to the Blynk app. If the water level exceeds a threshold, the buzzer and LED blink as an alert.
- The user can manually control the LED and buzzer via the Blynk app, overriding automatic functions if needed.
- The system continuously updates the sensor readings and responds based on predefined conditions and user inputs.

Chapter 3: Circuit and code

Sensors and Actuators

- **LDR (Light Dependent Resistor):**

The LDR detects ambient light levels and outputs an analog value based on the intensity. In low light conditions, the sensor triggers the LED to turn on automatically. The LDR reading is scaled and sent to the Blynk app for real-time monitoring.

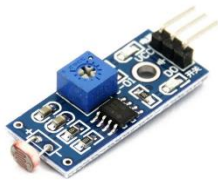


Fig – 1 LDR

- **Water Sensor (HW-038):**

The water sensor measures the presence of water or moisture and outputs an analog value. When the water level exceeds a set threshold, the LED blinks and the buzzer sounds, alerting the user of potential overflow. The sensor value is also displayed on the Blynk app.



Fig -2 Water sensor

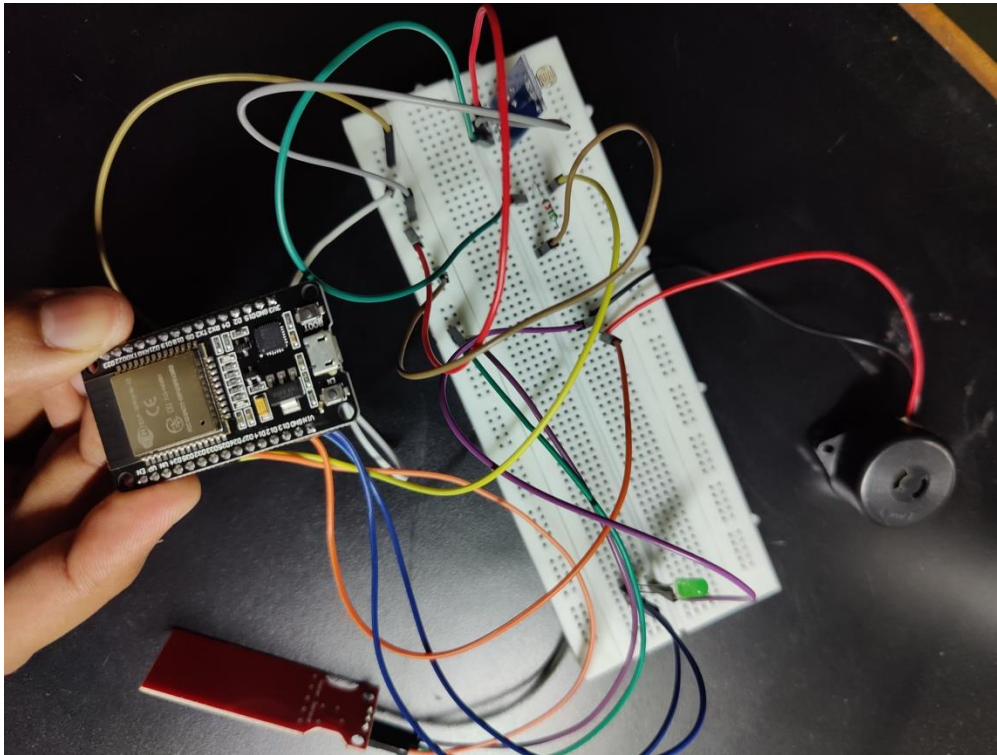
- **LED and Buzzer:**

The LED provides visual feedback, and the buzzer generates an audible alert. They are controlled automatically based on sensor readings and can also be operated manually via the Blynk app. This dual control enhances user flexibility and safety.



Fig – 3 Led & Buzzer

Circuit



Code

```
#define BLYNK_TEMPLATE_ID "TMPL3T-ACrkDD"
#define BLYNK_TEMPLATE_NAME "IOTPROJECT2"
#define BLYNK_AUTH_TOKEN "oT434sslDvn61k1lnkXgrCVYjd17c35S"
#define vpinLED V0
#define vpinBuzzer V1
#define vpinLDR V3
#define vpinWater V4

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

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "InternetByHarsh";
char pass[] = "helloharsh";

// Pin Definitions
const int ledPin = 13;
const int buzzerPin = 12;
const int ldrPin = 34;
const int waterPin = 35;

int ldrValue;
int scaledLdrValue;
int waterLevel;
```

```

int ledState = 0;
int buzzerState = 0;

BlynkTimer timer;

BLYNK_CONNECTED() {
  Blynk.syncVirtual(vpinLED);
  Blynk.syncVirtual(vpinBuzzer);
}

BLYNK_WRITE(vpinLED) {
  ledState = param.asInt();
  digitalWrite(ledPin, ledState ? HIGH : LOW);
  Serial.print("LED State: ");
  Serial.println(ledState ? "ON" : "OFF");
}

BLYNK_WRITE(vpinBuzzer) {
  buzzerState = param.asInt();
  digitalWrite(buzzerPin, buzzerState ? HIGH : LOW);
  Serial.print("Buzzer State: ");
  Serial.println(buzzerState ? "ON" : "OFF");
}

// LDR Sensor
void readLDR() {
  ldrValue = analogRead(ldrPin);
  scaledLdrValue = map(ldrValue, 0, 225, 100, 0);

  Blynk.virtualWrite(vpinLDR, scaledLdrValue);

  Serial.print("LDR Value: ");
  Serial.println(scaledLdrValue);

  // If scaled LDR value is below 20, turn on LED automatically
  if (scaledLdrValue < 20) {
    digitalWrite(ledPin, HIGH);
    Serial.println("Low light detected: LED turned ON automatically.");
  } else if (!ledState) {
    digitalWrite(ledPin, LOW);
  }
}

// Water Level Sensor
void readWaterSensor() {
  waterLevel = analogRead(waterPin);

  Blynk.virtualWrite(vpinWater, waterLevel);

  Serial.print("Water Level: ");
  Serial.println(waterLevel);
}

```



```

// If water level exceeds 1000, blink LED and buzzer
if (waterLevel > 1000) {
  Serial.println("High water level detected: Blinking LED and buzzing.");
  digitalWrite(ledPin, HIGH);
  digitalWrite(buzzerPin, HIGH);
  delay(200);
  digitalWrite(ledPin, LOW);
  digitalWrite(buzzerPin, LOW);
  delay(200);
}
}

void setup() {
  Serial.begin(115200);
  delay(1000);

  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ldrPin, INPUT);
  pinMode(waterPin, INPUT);

  digitalWrite(ledPin, LOW);
  digitalWrite(buzzerPin, LOW);

  Blynk.begin(auth, ssid, pass);

  timer.setInterval(1000L, readLDR);
  timer.setInterval(1000L, readWaterSensor);

  Serial.println("Welcome to our Home Automation System");
}

void loop() {
  Blynk.run();
  timer.run();
}

```

Explanation

The code initializes WiFi and connects to the Blynk platform for real-time monitoring and control. Sensor values are read and processed every second using timers. The LDR reading is scaled from 0-225 to 0-100, and if the value falls below 20, the LED turns on automatically. If the water sensor detects a reading above 1000, both the LED and buzzer activate to alert the user. The Blynk app allows manual control of the LED and buzzer, overriding automatic functions when needed.

Chapter 5: Results and Conclusion

The implemented Home Automation System using the ESP32 microcontroller successfully met all design requirements and functioned as intended. The results of the project are as follows:

- **LDR Sensor:**

The LDR sensor accurately detected changes in ambient light levels. The scaled output ranged from **0** to **100**, where **0** represented no light (dark environment) and **100** indicated maximum light (bright environment). The automatic LED control based on light intensity worked seamlessly, turning on the LED when the light level fell below the threshold of **20**.

- **Water Sensor:**

The water sensor reliably monitored moisture levels, providing real-time readings on the Blynk app. When the sensor value exceeded **1000**, the system activated an alert sequence: the LED blinked, and the buzzer emitted a sound, effectively notifying the user of high water levels.

- **LED and Buzzer Control:**

The LED and buzzer responded promptly to both automatic triggers from sensor readings and manual controls via the Blynk app. The dual-mode control provided flexibility and ensured user intervention could override automatic functions if needed.

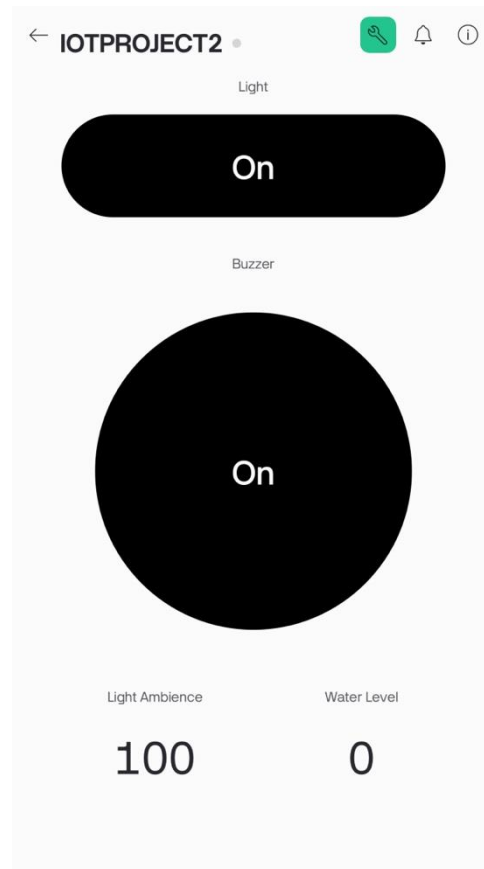
- **Blynk Integration:**

The Blynk app displayed real-time data from the LDR and water sensor, allowing the user to monitor sensor readings remotely. The app's buttons successfully controlled the LED and buzzer, providing a user-friendly interface for manual operation.

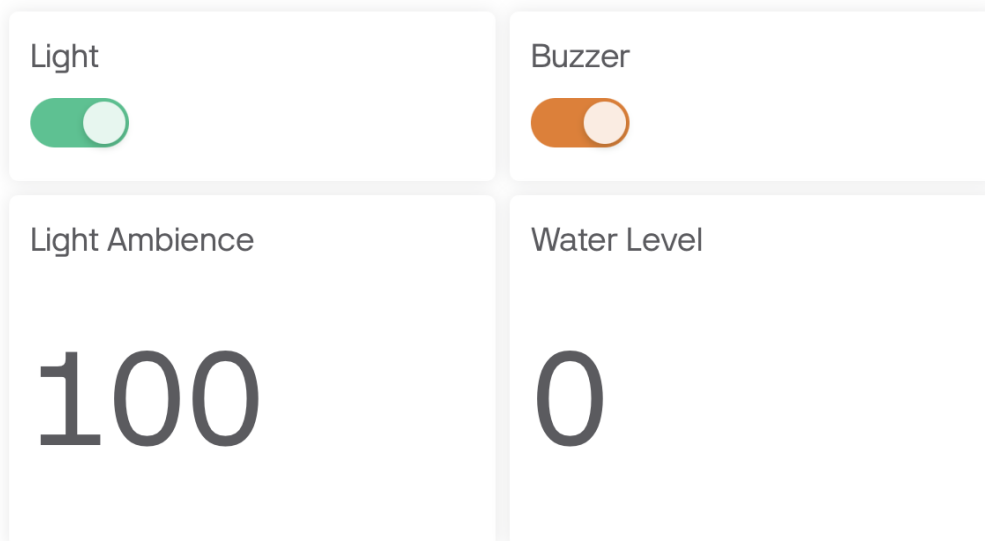
Overall, the project achieved its objectives of real-time monitoring, automatic control based on sensor inputs, and remote operation via the Blynk app. The system demonstrated stability, responsiveness, and accurate sensor readings under various environmental conditions.

Dashboards

Mobile UI



Web ui



Conclusion

The IoT-based Home Automation System using the ESP32 microcontroller was successfully designed, implemented, and tested. The project demonstrated the effective integration of sensors, actuators, and the Blynk platform, achieving seamless real-time monitoring and control. The LDR sensor provided accurate ambient light detection, enabling automatic LED control to optimize energy usage. The water sensor effectively monitored moisture levels, alerting users to potential overflow conditions with visual and audible signals. The Blynk app offered a user-friendly interface for remote control, enhancing convenience and accessibility. Overall, the system met its objectives of increasing safety, energy efficiency, and user flexibility, showcasing the potential of IoT solutions in smart home applications.