PROJECT TITLE:

IOT ENABLED SMART PACKAGING SYSTEM USING ESP32 or (SMART BAG)

Project Type:

Embedded Systems / IoT-Based Smart Project

Project Objective:

To monitor and protect packages during transit by sensing motion, location, temperature, pressure, light, and gas presence, and display live data on the Blynk IoT dashboard.

Done on:
April 2025

ABSTRACT

The increasing demand for safe, reliable, and intelligent logistics has led to the innovation of smart packaging systems. This project presents the design and implementation of a Smart Packaging System using the ESP32 microcontroller, integrated with various sensors to monitor and secure the package in real-time. The system includes a BMP280 sensor for temperature and pressure monitoring, MQ2 gas sensor for detecting harmful gases, MPU6050 for acceleration and gyroscopic motion sensing, LDR sensor to detect open/close status of the package, RC522 RFID module for item-level identification, and a Neo 6M GPS module for location tracking. These modules are interfaced with the ESP32, and the data is transmitted wirelessly to a Blynk IoT dashboard, enabling users to monitor sensor values and receive alerts remotely via a smartphone.

The system is designed to improve package safety, environmental monitoring, and content validation during transit. It effectively detects conditions such as gas leaks, package tampering, unauthorized access, vibrations due to mishandling, and provides accurate location tracking. Real-time alerts are generated when any abnormal condition is detected, thus preventing damage or loss of goods. The project demonstrates a cost-effective, efficient, and scalable solution for modern supply chains and sensitive deliveries such as food, electronics, and medical items.

The results confirm the system's reliability, responsiveness, and ease of use. Future improvements may include solar power integration, GSM connectivity, and blockchain for enhanced data security and global scalability. This project lays the foundation for a smart and connected future in logistics and product packaging.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 GENERAL

1. In the current era of rapid technological advancement, traditional packaging has evolved into intelligent systems capable of providing more than just physical protection for goods. Smart packaging systems are an integration of embedded sensors, microcontrollers, and wireless communication to ensure real-time monitoring, tracking, and management of products during transportation or storage.

2. The Internet of Things (IoT) plays a pivotal role in this transformation. It allows objects, including packages, to communicate with users and systems through sensors and cloud-based platforms. This project focuses on designing a smart packaging system that incorporates multiple environmental and safety sensors interfaced with an ESP32 microcontroller and controlled via the Blynk IoT platform.

3. This innovation can be applied in logistics, supply chain management, food and medical transportation, and any sector requiring real-time package monitoring. The proposed system is compact, energy-efficient, and cost-effective, offering significant benefits in terms of safety and traceability.

1.2 PROBLEM DEFINITION

Traditional packaging systems do not provide real-time feedback about the state of the contents or environmental conditions. This becomes a major issue in the transportation of sensitive goods such as electronics, pharmaceuticals, food, and confidential documents. Problems such as:

- Unauthorized opening or tampering of packages.
- Exposure to harmful gases or temperature fluctuations.
- Misplacement or theft of packaged items.
- Lack of real-time updates for logistics providers and customers.

Therefore, there is a critical need for a smart system that monitors multiple physical parameters, detects abnormalities, and sends alerts in real time. This project addresses these challenges by using a smart packaging solution built around an ESP32 controller and multiple sensors.

1.3 TECHNOLOGIES USED

The proposed system uses a combination of modern technologies to achieve smart monitoring capabilities:

- ESP32 Microcontroller: A powerful dual-core microcontroller with built-in Wi-Fi and Bluetooth used as the central processing and communication unit.

- Blynk IoT Platform: Enables real-time monitoring and control from a smartphone.

- GPS Module (NEO-6M): Used for real-time location tracking of the package.

- MPU6050 Sensor: Detects acceleration, vibration, and orientation, useful for motion or mishandling detection.

- BMP280 Sensor: Measures temperature and pressure to track environmental conditions.

- MQ2 Gas Sensor: Detects the presence of gases like smoke, LPG, methane, and hydrogen.

- LDR (Light Dependent Resistor): Senses whether the package is opened or closed based on light intensity.

- RC522 RFID Module: Used to check if specific items are present inside the package using RFID tags.

1.4 OBJECTIVES

The main objectives of the Smart Packaging System project are:

- To develop an IoT-based solution that ensures the safety and security of packaged goods.
- To measure and monitor key environmental parameters such as temperature, pressure, gas levels, and motion.
- To detect the open/close status of the package using an LDR sensor.
- To track the real-time location of the package using GPS.
- To identify the presence of tagged items using RFID technology.
- To display all data on a user-friendly dashboard (Blynk) in real-time.
- To alert users instantly in case of abnormalities or tampering.

1.5 ORGANISATION OF PROJECT REPORT

- This report is structured as follows:
- Chapter 1 introduces the project background, problem statement, and technologies used.
- Chapter 2 reviews existing systems and related literature to understand current trends and gaps.
- Chapter 3 explains the methodology, including software and hardware implementation, block diagrams, and circuit explanations.
- Chapter 4 presents the results, observations, performance analysis, and applications.
- Chapter 5 concludes the project with key findings and future scope.
- Chapter 6 includes references used throughout the research and development process.

1.6 CONCLUSION

The introduction chapter establishes the need for a smart packaging system that enhances traditional packaging using embedded electronics and IoT. It outlines the technologies involved and the main goals of the project. The proposed solution aims to provide users with peace of mind by offering transparency, safety, and control over their packages during transit. With this foundation, the project moves forward into a detailed review of related work in Chapter 2.

CHAPTER 2
LITERATURE REVIEW

2.1 INTRODUCTION

The field of smart packaging has gained significant attention over the last decade due to the growing demand for real-time product tracking, security, and environmental condition monitoring during transit. Various research efforts have explored the integration of sensors, microcontrollers, and wireless communication for intelligent packaging. This chapter presents an overview of existing systems, techniques, and technologies, identifying their advantages and limitations, and justifying the need for the proposed system.

2.2 EXISTING SYSTEMS

1. Traditional Packaging

Traditional packaging serves the purpose of enclosing and protecting products. However, it lacks intelligent features such as monitoring or feedback. It is purely passive and depends heavily on human intervention for tracking and inspection.

2. RFID-Based Tracking Systems

Radio Frequency Identification (RFID) is widely used for inventory tracking. Some studies have shown the use of RFID tags to detect the presence of items within a package. However, it doesn't offer environmental sensing or real-time alerts in case of tampering or abnormal conditions.

3. GSM/GPS-Based Tracking

Some researchers proposed GPS-enabled packages for location tracking combined with GSM-based SMS alerts. While effective for tracking, these systems often consume high power and may lack additional safety or environmental sensing features.

4. Smart Food Packaging

Studies in food safety have integrated sensors to monitor temperature and humidity inside food containers. These are useful in cold chains but are often limited to only one or two parameters and lack real-time alert systems or mobile integration.

5. Commercial Solutions

Companies like Amazon and FedEx have experimented with smart labels and sensors for tracking. However, these are proprietary, expensive, and inaccessible to smaller businesses or academic use.

2.3 REVIEW OF RESEARCH PAPERS AND JOURNALS

1. "IoT-Based Smart Parcel Monitoring System" – IEEE (2021)

- Describes an IoT system using GPS, temperature, and humidity sensors.
- Limited in detecting vibrations or unauthorized opening.

- Lacked a user interface for real-time dashboard viewing.

2. "Smart Packaging Using Arduino and GSM" – IJERT (2020)

- Implemented basic gas and temperature sensing.
- Used Arduino Uno with GSM but had no GPS or mobile app integration.

3. "Secure Packaging System Using RFID and Buzzer Alert" – IJRASET (2019)

- Focused on item authentication using RFID.
- No environmental sensors or real-time tracking included.

4. "Cold Chain Logistics Monitoring using IoT" – Springer (2022)

- Efficient for perishable goods, used temperature sensors and cloud monitoring.
- Did not address theft, vibrations, or open-package alerts.

2.4 GAP IDENTIFICATION

- From the above studies and systems, the following gaps are observed:

- Lack of integrated systems that combine environmental monitoring, motion detection, item tracking, and location monitoring.

- Few solutions offer a mobile app dashboard with real-time sensor data.

- Most research projects are limited in scope and address only one or two packaging concerns.

- Commercial systems are costly and inaccessible for learning or customization.

2.5 NEED FOR THE PROPOSED SYSTEM

- To address the identified gaps, the Smart Packaging System is proposed with the following advantages:

- Multi-sensor Integration: Combines gas, pressure, temperature, vibration, motion, light, and item tracking sensors.

- Real-time Dashboard: Uses the Blynk IoT platform to provide a mobile interface for live data and alerts.

- Security Features: Detects if a package is opened, dropped, or tampered with.

- Affordable and Scalable: Based on ESP32 and open-source tools, making it ideal for academic, commercial, and industrial use.

2.6 CONCLUSION

This literature review reveals that while there are multiple approaches to smart packaging, none of the existing systems offer a complete and cost-effective solution with comprehensive sensor integration and real-time mobile monitoring. The proposed system bridges this gap by combining critical features needed for secure, efficient, and smart packaging.

CHAPTER 3
METHODOLOGY

3.1 INTRODUCTION

This chapter describes the systematic approach adopted to design and implement the Smart Packaging System. It includes both software and hardware methodologies for building a prototype capable of monitoring package conditions like temperature, pressure, vibrations, gas leakage, location, and tampering. The system architecture, development tools, and implementation stages are explained in detail to provide a clear understanding of the proposed solution.

3.2 SOFTWARE IMPLEMENTATION

The software is developed using the Arduino IDE and Blynk IoT platform. The ESP32 microcontroller acts as the brain of the system, collecting data from sensors and sending it to the Blynk mobile dashboard through Wi-Fi.

Tools Used:

- Arduino IDE: For programming ESP32 in C/C++.

- Blynk IoT Platform: For real-time sensor data monitoring on mobile.

- Libraries Used:
- Wire.h (I2C communication)
- Adafruit_Sensor, Adafruit_BMP280 (for BMP280)
- MPU6050.h (for accelerometer and gyroscope)
- MQ2.h (for gas sensor)
- SPI.h and MFRC522.h (for RFID)
- BlynkSimpleEsp32.h (for Blynk integration)

Data streams in Blynk:

- V0: Temperature (BMP280)
- V1: Pressure (BMP280)
- V2: Gas Levels (MQ2)
- V3–V5: Accelerometer (X, Y, Z)

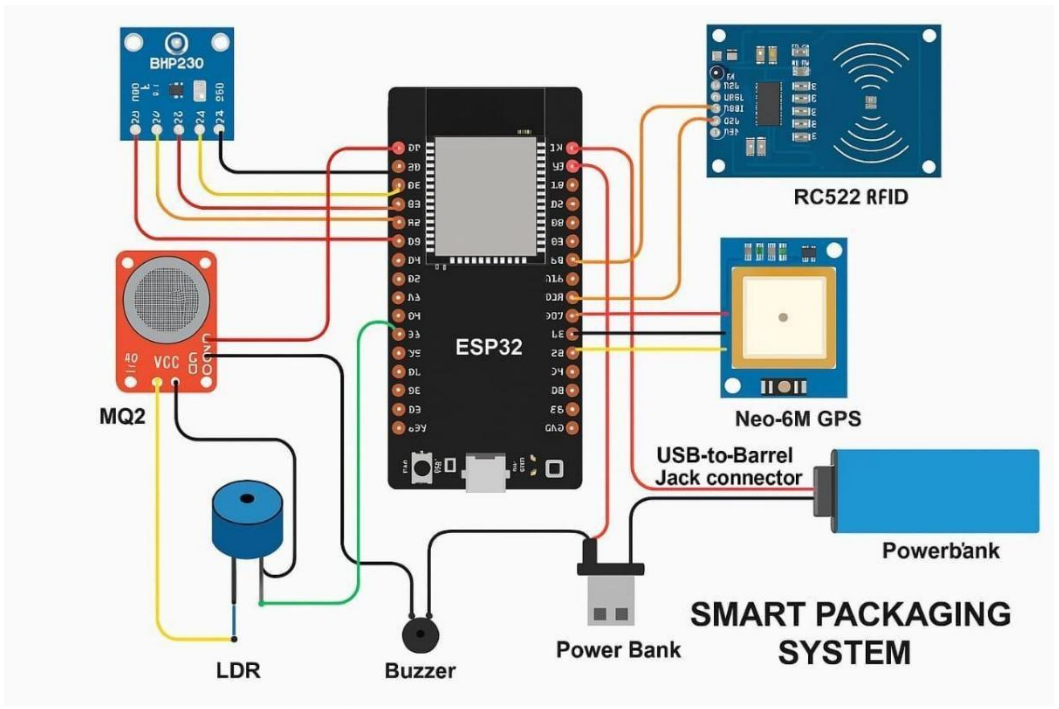
- V6–V8: Gyroscope (X, Y, Z)

- V9: LDR (Open/Close detection)

3.3 PROTOTYPE OVERVIEW

The prototype of the smart packaging system integrates several sensors and modules on the ESP32 board. The sensors continuously monitor physical parameters and transmit them to the Blynk IoT dashboard

3.3.1 CIRCUIT DIAGRAM



Description:

- ESP32: Central controller for processing and Wi-Fi.
- BMP280: Measures temperature and pressure.
- MQ2: Detects smoke, LPG, and gas leaks.
- MPU6050: Monitors acceleration, vibrations, and tilt.
- LDR: Detects package open/close using light exposure.
- Neo 6M GPS: Tracks location.
- RC522 RFID: Identifies items inside the package.
- Blynk App: Displays data remotely via smartphone.

3.4 HARDWARE IMPLEMENTATION

This section covers the physical connection of sensors and modules to the ESP32 DevKit V1.

Hardware Components:

- ESP32 Devkit V

- BMP280 (I2C: GPIO21 - SDA, GPIO22 - SCL)
- MPU6050 (I2C: Same as above)
- MQ2 (Analog: GPIO34)
- LDR (Analog: GPIO35)
- Neo6M GPS (UART: GPIO16 - RX, GPIO17 - TX)
- RC522 RFID (SPI: GPIO5 - SCK, GPIO23 - MOSI, GPIO19 - MISO, GPIO18 - SS, GPIO2 - RST)
- Power Supply: 3.3V (ESP32), 5V for some modules

3.5 WORKING SUMMARY

- When the package is closed and sealed, all sensors are active.
- Any unusual movement (fall or vibration), temperature/pressure changes, gas detection, or package opening triggers an update to Blynk.
- RFID tags ensure that the correct items are placed inside. Any mismatch raises an alert.
- GPS module continuously updates the location.

- All data is visible on the Blynk mobile app in real-time, allowing users to monitor package safety and environmental conditions.

3.6 BUZZER MODULE

3.6.1 Working Principle of Buzzer

A buzzer is an audio signaling device that emits a beep sound when voltage is applied. It typically works based on the piezoelectric principle, converting electrical signals into mechanical vibrations, which generate sound. In this system, it acts as an alert system triggered remotely.

3.6.2 Circuit Diagram of Buzzer with ESP32

The buzzer's positive pin is connected to GPIO25 of the ESP32, and the negative pin is grounded. The circuit is simple and can be powered directly through the 3.3V logic.

```
[ESP32 GPIO25]---- > [Positive Pin of Buzzer]
[ESP32 GND]-------> [Negative Pin of Buzzer]
```

3 6.3 Blynk Control using Virtual Pin V10

In the Blynk IoT dashboard, a button widget is linked to V10 (virtual pin). When toggled ON, it sends a signal to ESP32 to activate the buzzer by making GPIO25 HIGH; when OFF, it sets it to LOW. This enables remote audible alerts.

CHAPTER 4
RESULTS

4.1 INTRODUCTION

This chapter presents the outcomes of implementing the Smart Packaging System. Each module was tested individually and then integrated to form a complete system. The results are displayed using the Blynk IoT dashboard, and the system's responses to various simulated conditions are recorded and analyzed.

4.2 INDIVIDUAL MODULE OUTPUTS

1. BMP280 Sensor (Temperature and Pressure Monitoring)

- Setup: Connected via I2C (GPIO21 & GPIO22).
- Result: Displayed real-time temperature and pressure on the Blynk app.
- Sample Output:
- Temperature: 28.5°C
- Pressure: 1008.3 hPa
- Status: Accurate and stable readings under different environmental conditions.

2. MQ2 Gas Sensor

- Setup: Connected to analog pin GPIO34.
- Test: Brought lighter flame nearby (without ignition) to simulate gas leakage.
- Result: Detected increased gas levels; Blynk V2 value spiked

- Sample Output:
- Normal: ~120 (safe)
- Gas Detected: ~450+ (alert)

- Status: Prompt and reliable detection of combustible gases.

3. MPU6050 Sensor (Accelerometer and Gyroscope)

- Setup: I2C pins GPIO21 and GPIO22.
- Test: Shook or tilted the package to simulate vibration or fall.
- Result:
- V3–V5 (Accel X, Y, Z): Fluctuated based on movement
- V6–V8 (Gyro X, Y, Z): Showed angular displacement

- Status: Movement detection is fast and sensitive to minor shifts.

4. LDR Sensor (Package Open/Close Detection)

- Setup: Connected to analog pin GPIO35.
- Test: Package lid was opened and closed under varying light.
- Result:
- Open: Light detected → LDR value increased

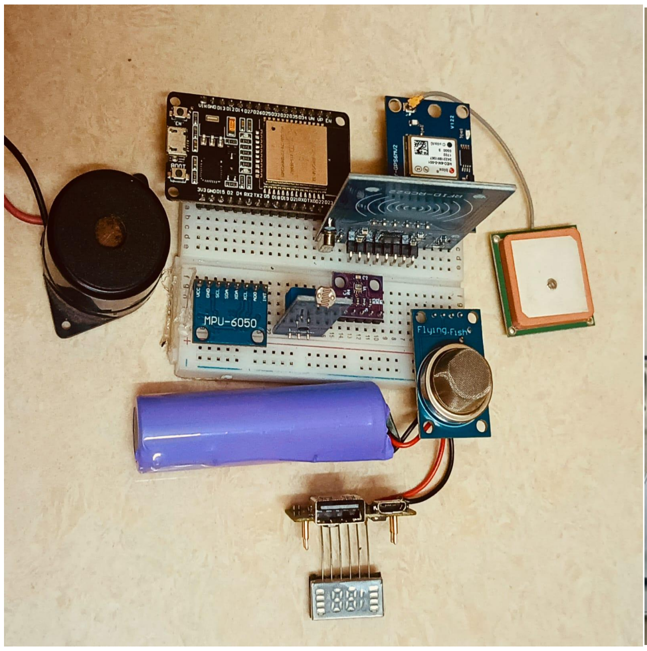
- Closed: Darkness → LDR value decreased
- Status: Accurately detects tampering or unauthorized access.

5. RC522 RFID Module

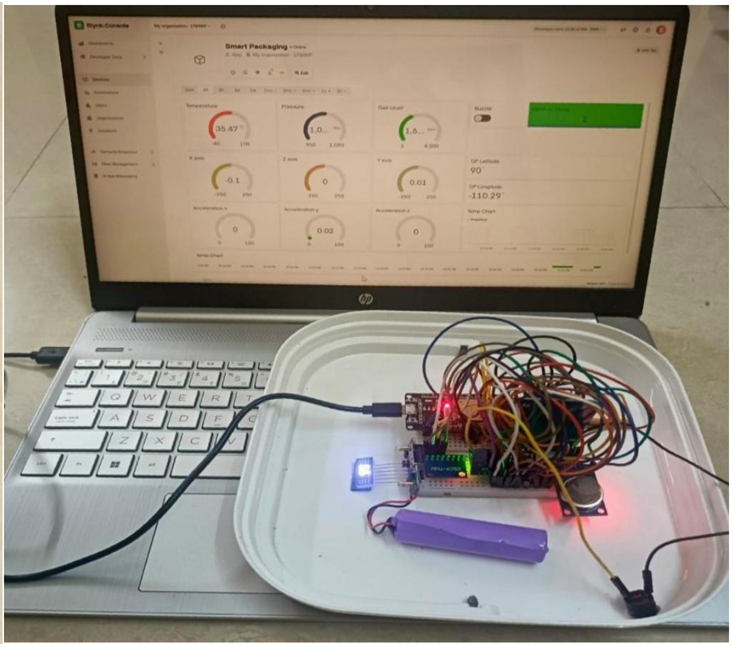
- Setup: SPI (GPIO5, GPIO23, GPIO19, GPIO18, GPIO2).
- Test: Scanned a pre-registered RFID tag.
- Result:
- Correct Tag: "Item verified"
- Unknown Tag: "Item mismatch – alert!"
- Status: Efficient item validation system.

6. Neo 6M GPS Module

- Setup: UART (GPIO16 & GPIO17).
- Test: Device was moved to different locations.
- Result: Displayed updated latitude and longitude values.
- Sample Output:
- Latitude: 13.0500, Longitude: 80.2110
- Status: Real-time tracking works well outdoors; GPS fix may take a few seconds.



Component Image



Project Image

4.3 SYSTEM INTEGRATION RESULTS

When all sensors and modules were integrated:

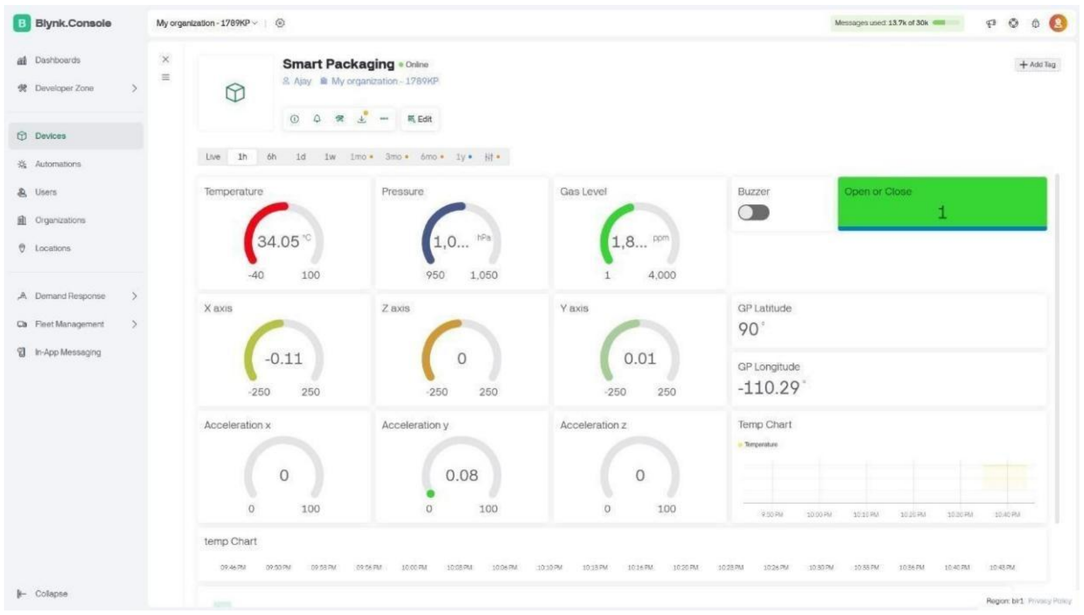
- All sensor data was updated simultaneously in the Blynk dashboard.
- No interference or lag was observed among sensor readings.
- Real-time alerts were triggered in the Blynk app when threshold values were crossed.

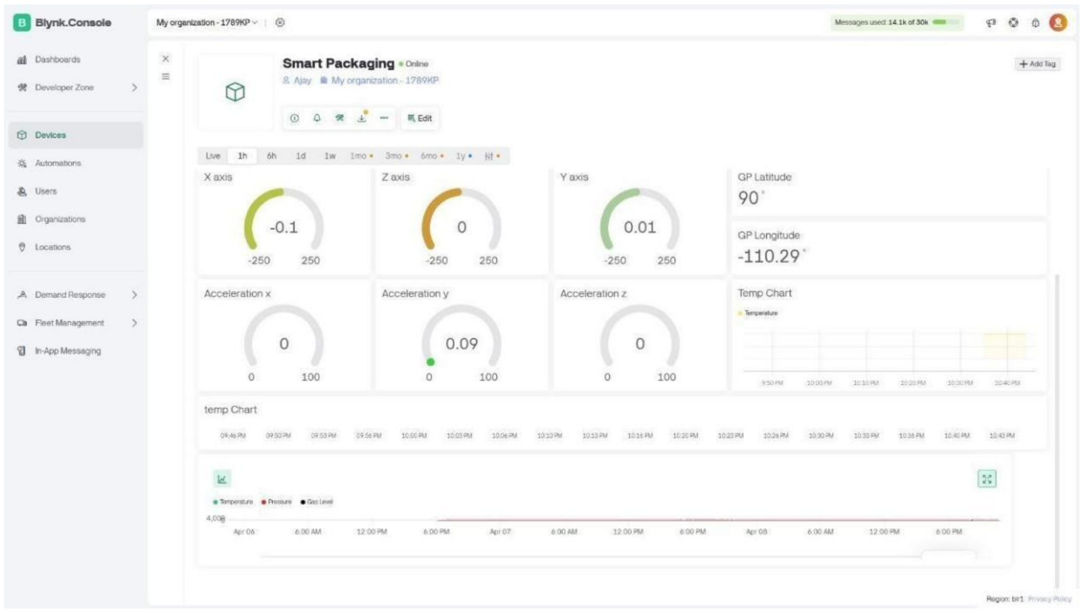
Snapshot of Live Dashboard (Blynk App):

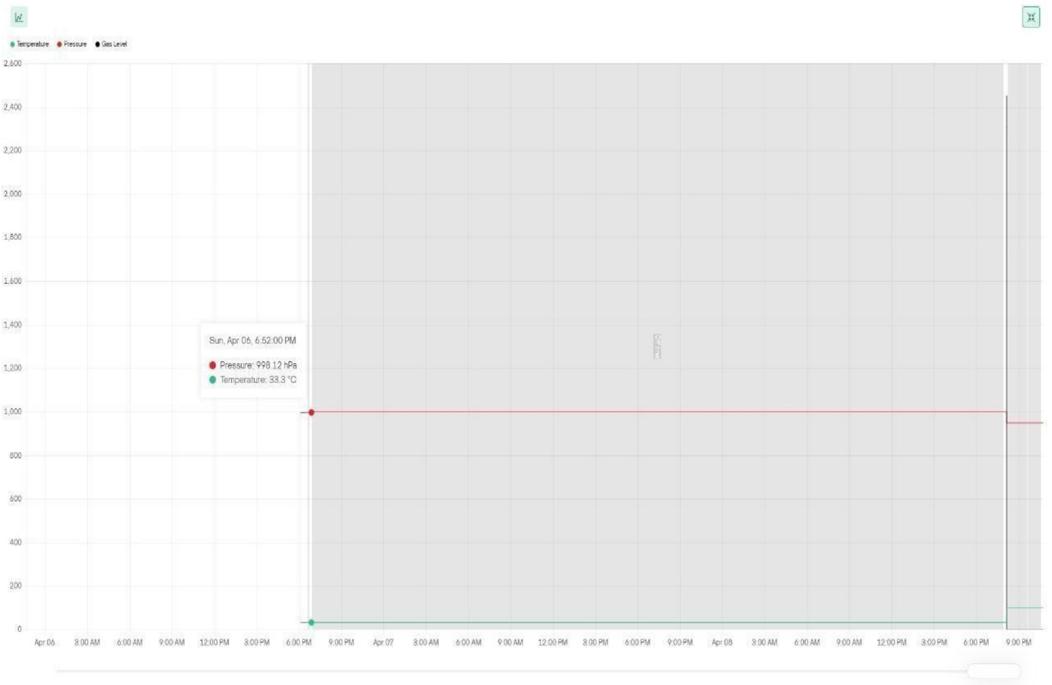
- V0: 29°C
- V1: 1005 hPa
- V2: 300 (Gas safe)
- V3–V5: [0.01, 0.02, 0.00]

- V6–V8: [0.1, -0.05, 0.0]
- V9: 50 (Package closed)

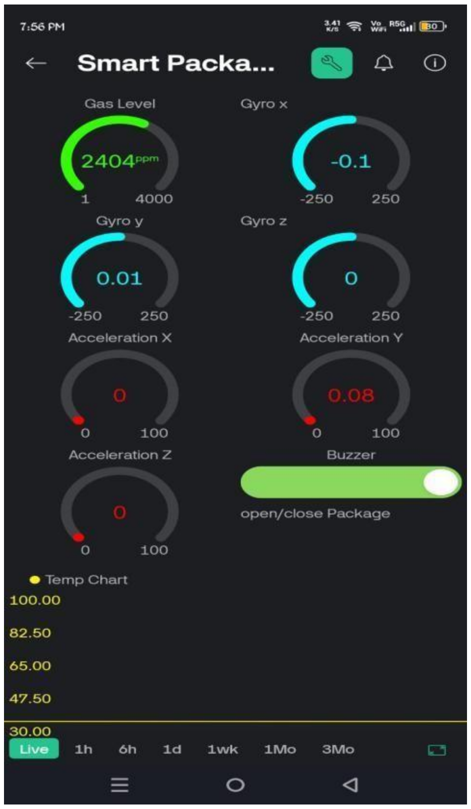
4.3.1 LAPTOP DASHBOARD

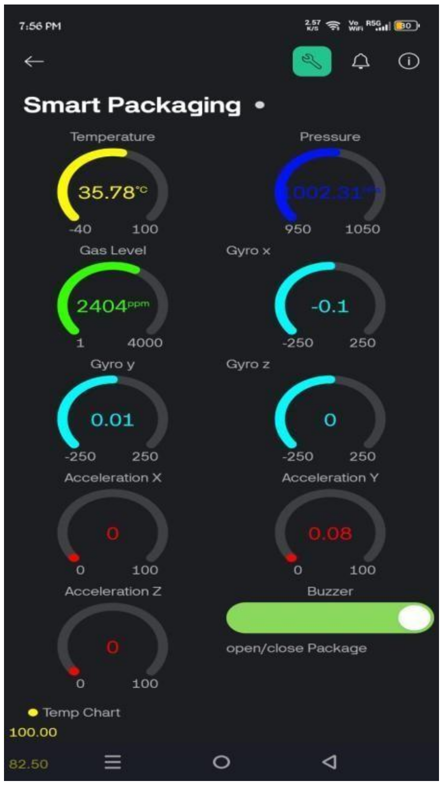



Blynk.Console
My organization - 1789KP
Messages used: 14.1k of 30k
Dashboards
Developer Zone
Devices
Automations
Users
Organizations
Locations
Demand Response
Fleet Management
In-App Messaging
Smart Packaging
Online
Ajay
My organization - 1789KP
Edit
Add Tag
Live
1h
6h
1d
1w
1mo
3mo
6mo
1y
X axis
-0.1
Z axis
0
Y axis
0.01
GP Latitude
90°
GP Longitude
-110.29°
Acceleration x
0
Acceleration y
0.09
Acceleration z
0
Temp Chart
Temperature
temp Chart
Temperature
Pressure
Gas Level
Region: blr1
Privacy Policy

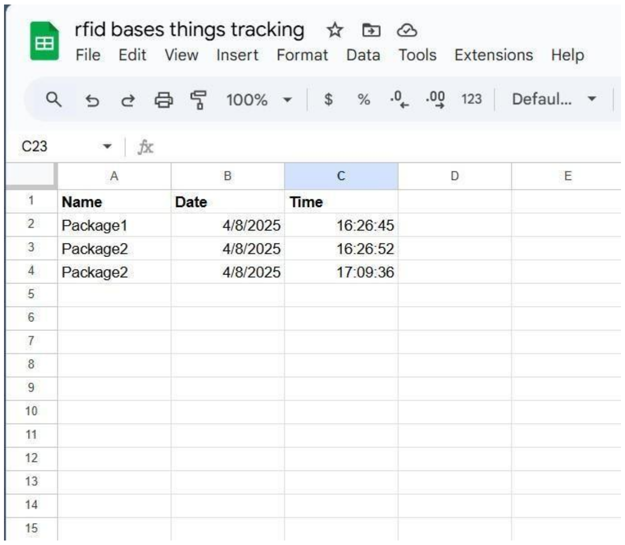

Temperature
Pressure
Gas Level
Sun, Apr 06, 6:52:00 PM
Pressure: 998.12 hPa
Temperature: 33.3 °C
Apr 06
Apr 07
Apr 08

4.3.2 MOBILE DASHBOARD





4.3.3 RFID DATABASE THINGS TRACKING



Name	Date	Time
Package1	4/8/2025	16:26:45
Package2	4/8/2025	16:26:52
Package2	4/8/2025	17:09:36

4.4 ALERT SYSTEM AND THRESHOLD BEHAVIOR

SENSORS	**CONDITION**	**ACTION**
MQ2	>400	Gas alert triggered
LDR	>200	Package Open alert
MPU6050	Movement detected	Vibration Alert
RFID	Wrong tag	Item Mismatch Alert

4.5 ADVANTAGES OBSERVED

- Real-Time Monitoring: All sensor values are visible instantly via mobile.
- Portability: Lightweight and compact, suitable for various packages.
- Cost-Effective: Uses widely available components.
- Security: Unauthorized access and tampering are detected instantly.

4.6 LIMITATIONS

- GPS requires open sky for best accuracy.

- Blynk requires stable internet connection.
- Power consumption is moderate; needs battery backup for extended use.

4.7 BUZZER CODE INTEGRATION VIA BLYNK

4.7.1 Virtual Pin Configuration

In the Blynk IoT platform, a virtual pin (V10) is assigned to control the buzzer. The toggle button on the dashboard sends a digital value (1 or 0) to the ESP32 through the internet.

4.7.2 Code Explanation

Below is the snippet added to the main Arduino code:

```
  #define BUZZER_PIN 25  // GPIO pin connected to the buzzer

  BLYNK_WRITE(V10) {
 int buzzerState = param.asInt(); // 1 = ON, 0 = OFF
 digitalWrite(BUZZER_PIN, buzzerState);
}

 void setup()
 { pinMode(BUZZER_PIN, OUTPUT);
 digitalWrite(BUZZER_PIN, LOW);  // Ensure buzzer is off initially

 // Existing setup code for Blynk, sensors, etc.
 }
```

4.6.3 Logic Description

BLYNK_WRITE(V10) is a callback function triggered when the Blynk app's toggle switch changes state.

The state (1 or 0) is read using param.asInt().

digitalWrite() controls the buzzer based on this value.

4.8 SUMMARY

The Smart Packaging System successfully detects and reports environmental and physical conditions of a package in real-time. Each module performed as expected during individual and integrated testing phases. The system proved to be reliable, scalable, and highly beneficial for secure and intelligent logistics.

CHAPTER 5
CONCLUSION

5.1 CONCLUSION

The Smart Packaging System was successfully designed, developed, and tested to meet its intended objectives of enhancing package security, traceability, and environmental monitoring. By integrating multiple sensors—such as BMP280, MQ2, MPU6050, LDR, RC522 RFID, and a GPS module—with the ESP32 microcontroller and Blynk IoT platform, the system offers a real-time, remote solution for intelligent logistics.

The solution effectively detects temperature, pressure, gas leakage, movement/vibration, and tampering of packages. RFID ensures item-level verification, while GPS provides live location updates. The Blynk app enables users to receive instant alerts and visualize data from anywhere in the world. The prototype confirms that smart packaging can significantly reduce losses, delays, and damage, especially in sensitive supply chains such as pharmaceuticals, food delivery, and electronics.

5.2 ACHIEVEMENTS

- Real-time monitoring and control using IoT.
- Accurate environmental sensing with low-cost components.
- Integration of location tracking with security features.
- Seamless mobile dashboard with alert notifications.
- RFID-based validation for contents inside the package.
- Successfully tested under multiple practical conditions.

5.3 LIMITATIONS

- Power dependency: Requires a continuous power supply or battery backup.
- GPS limitations: Reduced accuracy indoors or in obstructed environments.
- Internet requirement: Blynk dashboard requires internet access for communication.
- Data privacy: Location and sensor data transmission must be securely encrypted in a commercial deployment.

5.4 BUZZER CONTROL RESULTS

5.4.1 Blynk App Interface

A toggle button was added to the Blynk dashboard linked to virtual pin V10. When the user switches the button ON, the buzzer connected to GPIO25 emits a sound, and it turns OFF when the button is toggled OFF.

Figure 5.4.1 – Blynk Dashboard Showing Buzzer Control Toggle
(Insert screenshot here)

5.4.2 Test Case: Remote Buzzer Activation

TEST SCENARIO	**ACTION PERFORMED**	**EXPECTED OUTPUT**	**RESULT**
Toggle ON In Blynk App	Button Pressed (VIO=1)	Buzzer Turns ON (beeps)	PASS

Toggle OFF in Blynk app	Button Released (VIO=0)	Buzzer Turns OFF (Silent)	PASS
Power Cycle and reconnect	Reconnect Blynk app	Buzzer Stays OFF by default	PASS

5.4.3 Discussion

The buzzer feature enhances the interactivity of the smart packaging system by providing an immediate audible alert upon user request. This can be useful for notifying handlers during critical conditions or for quick identification of a package.

5.5 FUTURE WORK

To make the system more robust, scalable, and commercially viable, the following improvements can be considered:

1. Solar-Powered System

Integrate a small solar panel and battery to ensure uninterrupted power supply during transit.

2. Local Data Storage

Add SD card module for offline data logging when Wi-Fi or mobile data is unavailable.

3. GSM/GPRS Module

Replace Wi-Fi with GSM (SIM800L/SIM900A) to enable data transmission without depending on local Wi-Fi.

4. Edge AI Integration

Use machine learning models to predict anomalies based on sensor patterns (e.g., frequent shocks could indicate damage in transit).

5. Tamper-Proof Enclosure

Design a secure casing that physically protects the hardware from manipulation or environmental exposure.

6. Blockchain Integration

Secure transaction logs and item tracking records using blockchain for transparent supply chain management.

7. Battery Health Monitoring

Implement battery management system (BMS) to monitor charge level, voltage, and life cycle.

8. Multi-Package Monitoring

Expand the system to simultaneously monitor multiple packages via a central hub or master-slave architecture.

5.6 SUMMARY

The Smart Packaging System is a proof-of-concept that shows how IoT can revolutionize logistics and delivery systems. While the current model is a prototype, its scope for commercial application is vast. With further refinement and enhancements, this system can greatly improve supply chain transparency, product safety, and customer satisfaction.

APPENDICES

APPENDIX A: ESP32 PIN MAPPING:

MODULE	**SENSOR**	**INTERFACE**	**ESP32 pin**
Environmental Sensor	BMP280	12C	SDA-GPIO21, SCL-GPIO22
Motion Sensor	MPU6050	I2C	SDA-GPI021, SCL-GPIO22
Gas Sensor	MQ2	Analog	A0-GPIO34
Light Sensor	LDR	Analog	A0-GPIO35
RFID Module	RC522	SPI	SDA - Module GPIO5, SCK - GPIO18, MOSI - GPIO23, MISO-GPIO19, RST - GPIO4

GPS Module	NEO 6M	UART	TX - GPIO16, RX - GPIO17
Power supply	All modules	VCC/GND	3.3V GND

APPENDIX B: ARDUINO LIBRARIES USED:

- Wire.h – For I2C communication.
- Adafruit_BMP280.h – For BMP280 sensor.
- Adafruit_Sensor.h – Unified sensor support.
- MPU6050.h – For motion sensing.
- SPI.h & MFRC522.h – For RFID module.
- TinyGPSPlus.h – For GPS data parsing.
- WiFi.h – For ESP32 Wi-Fi.
- BlynkSimpleEsp32.h – For Blynk dashboard communication.

APPENDIX C: SAMPLE CODE SNIPPET – SENSOR READING & BLYNK TRANSMISSION:

```
#define BLYNK_TEMPLATE_ID "TMPL3bKaPvTg2"
#define BLYNK_TEMPLATE_NAME "Smart Packaging"
#define BLYNK_AUTH_TOKEN "q0510MAFnnPfsRw0suk59KDdzwcXQZrP"
```

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <SPI.h>
#include <MFRC522.h>
#include <HTTPClient.h>
#include <WiFiClientSecure.h>
#include <TinyGPS++.h>

// WiFi credentials
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Airtel_Ajay &Satya";
char pass[] = "77777777";

// Sensor pins
#define MQ2_PIN 34
#define LDR_PIN 25
#define BUZZER_PIN 15
#define RST_PIN 22
#define SS_PIN 21
#define GPS_TX_PIN 17
#define GPS_RX_PIN 16

// RFID setup
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
MFRC522::StatusCode status;
int blockNum = 2;
```

```
byte bufferLen = 18;
byte readBlockData[18];
String card_holder_name;
const String sheet_url =
"https://script.google.com/macros/s/AKfycbwzSt3PE1WWMfZi9kU3JgyTtEXKx
9CA621t3ewH9J6wvPyNYywCcPPV81i4nwMBH1ZW/exec?name=";

// Sensor objects
Adafruit_MPU6050 mpu;
Adafruit_BMP280 bmp;
TinyGPSPlus gps;
float lat = 0, lng = 0;

// Buzzer controlled via Blynk toggle switch (V12)
BLYNK_WRITE(V12) {
  int value = param.asInt();
  digitalWrite(BUZZER_PIN, value);
}

void setup()
  { Serial.begin(115200);
  Wire.begin(21, 22);
  SPI.begin();
  Serial1.begin(9600, SERIAL_8N1, GPS_TX_PIN, GPS_RX_PIN);

  pinMode(LDR_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW); // Start with buzzer OFF

  Blynk.begin(auth, ssid, pass);

  if (!bmp.begin(0x76)) {
    Serial.println("BMP280 not found");
    while (1);
  }

  if (!mpu.begin())
    { Serial.println("MPU6050 not
    found"); while (1);
  }

  mfrc522.PCD_Init();
}

void loop()
  { Blynk.run();

  // Read Sensors
  int mq2Value = analogRead(MQ2_PIN);
  int ldrStatus = digitalRead(LDR_PIN);
  String packageStatus = (ldrStatus == 0) ? "1" : "0";
  float temp = bmp.readTemperature();
  float pressure = bmp.readPressure() / 100.0;

  sensors_event_t a, g, temp_event;
  mpu.getEvent(&a, &g, &temp_event);

  // Read GPS
  unsigned long start = millis();
   while (millis() - start < 2000)
    { while (Serial1.available())
    { gps.encode(Serial1.read());
    }
  }
```

```
if (gps.location.isValid())
  { lat = gps.location.lat();
  lng = gps.location.lng();
}

// Send data to Blynk
Blynk.virtualWrite(V2, temp);
Blynk.virtualWrite(V3, pressure);
Blynk.virtualWrite(V5, mq2Value);
Blynk.virtualWrite(V4, a.acceleration.x);
Blynk.virtualWrite(V10, a.acceleration.y);
Blynk.virtualWrite(V11, a.acceleration.z);
Blynk.virtualWrite(V7, g.gyro.x);
Blynk.virtualWrite(V8, g.gyro.y);
Blynk.virtualWrite(V9, g.gyro.z);
Blynk.virtualWrite(V6, packageStatus);
Blynk.virtualWrite(V0, lat);
Blynk.virtualWrite(V1, lng);

// RFID Section
if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial())
  { ReadDataFromBlock(blockNum, readBlockData);

  Serial.print("RFID Data: ");
  for (int j = 0; j < 16; j++) {
    Serial.write(readBlockData[j]);
  }
  Serial.println();

  if (WiFi.status() == WL_CONNECTED)
    { WiFiClientSecure client;
    client.setInsecure();
```

```
      card_holder_name = sheet_url + String((char*)readBlockData);
      card_holder_name.trim();

      HTTPClient https;
      if (https.begin(client, card_holder_name))
        { int httpCode = https.GET();
        if (httpCode > 0) {
          Serial.printf("[HTTPS] GET... code: %d\n", httpCode);
        } else {
          Serial.printf("[HTTPS] GET... failed, error: %s\n",
https.errorToString(httpCode).c_str());
        }
        https.end();
      } else {
        Serial.println("[HTTPS] Unable to connect");
      }
    }
  }

  delay(2000);
}

void ReadDataFromBlock(int blockNum, byte readBlockData[])
  { for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }

  status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
blockNum, &key, &(mfrc522.uid));
  if (status != MFRC522::STATUS_OK) {
    Serial.print("Authentication failed for Read: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

  status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);
  if (status != MFRC522::STATUS_OK) {
    Serial.print("Reading failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

  Serial.println("Block was read successfully");
}
```

REFERENCE

1. ESP32 DevKit V1 Datasheet – Espressif Systems
https://www.espressif.com/en/products/socs/esp32

2. BMP280 Sensor Datasheet – Bosch Sensortec
https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/

3. MQ2 Gas Sensor Datasheet – Hanwei Electronics
https://www.componentsinfo.com/mq2-gas-sensor/

4. MPU6050 Sensor Datasheet – InvenSense
https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/

5. RC522 RFID Module Datasheet – NXP Semiconductors
https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf

6. Neo 6M GPS Module Datasheet – u-blox
https://www.u-blox.com/en/product/neo-6-series

7. Blynk IoT Platform
https://blynk.io

8. Arduino IDE Software
https://www.arduino.cc/en/software

9. Adafruit Unified Sensor Library for BMP280 – GitHub
https://github.com/adafruit/Adafruit_BMP280_Library

10. MPU6050 Arduino Library – Jeff Rowberg
https://github.com/jrowberg/i2cdevlib

11. RFID RC522 Library for Arduino – Miguel Balboa
https://github.com/miguelbalboa/rfid

12. Technical articles and community discussions from:

- Electronics Hub
- Circuit Digest
- Random Nerd Tutorials
- Stack Overflow

13. IEEE Research Articles on Smart Packaging and IoT Applications.