International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)
ISSN: 2456-236X
Vol. **11** Special Issue **03** | **2026**

Bluetooth-Enabled Smart Medicine Box

Mr. Shreedhar Jayesh Pawar [1], Mrs. Laxmi Subhash Sawant[2], Mrs. Shweta Santosh Sutar[3], Mr. Akshay Arun Parulekar [4], Mrs. Tanvi Rahul Singhan [5]

[1,2,3,4,5]*Assistant Professor, Bsc.IT/CS Department, MITM College Sukalwad- Sindhudurg, Maharashtra, India*

DOI: 10.5281/zenodo.20630277

ABSTRACT

Medication non-adherence is a major challenge, particularly for elderly patients and individuals with chronic illnesses. This research paper presents the design and implementation of a Bluetooth-enabled smart medicine box that autonomously opens specific medicine drawers at scheduled times. The system uses a Flutter mobile application for user interaction and scheduling, and an ESP-based IoT device (ESP32) for local execution and control. The Flutter app communicates with the medicine box using Bluetooth and consumes REST APIs exposed by the ESP32 using the Dio HTTP client. Once schedules are transferred, the ESP32 stores them locally and executes them autonomously, ensuring reliable operation even when the mobile application is disconnected. The proposed system is low-cost, offline-capable, and suitable for home healthcare environments.

Keywords
Smart Medicine Box, ESP32, Flutter, Bluetooth Communication, REST API, Medication Adherence

1. INTRODUCTION

Medication adherence plays a critical role in effective healthcare management. Traditional pill organizers rely on manual reminders and are prone to human error. With advancements in IoT and mobile technologies, automated medicine dispensing systems have become feasible. This paper proposes a smart medicine box system that integrates a Flutter-based mobile application with an ESP32-based embedded system to provide accurate, automated, and user-friendly medicine dispensing.
Unlike cloud-dependent solutions, the proposed system emphasizes local communication using Bluetooth and on-device scheduling, making it robust against internet unavailability. The system ensures that only the correct medicine drawer opens at the prescribed time, reducing the risk of missed or incorrect dosages.

2. SYSTEM ARCHITECTURE

The system consists of two main components:
Flutter Mobile Application
ESP32-based Smart Medicine Box

2.1 Architecture Overview

The overall architecture is illustrated conceptually as follows:
Flutter App → Bluetooth → ESP32 REST API → Scheduler → Motor Control → Drawer Mechanism
The Flutter application is responsible for user interaction and schedule configuration, while the ESP32 device handles local storage, scheduling, and physical drawer control.

3. FLUTTER MOBILE APPLICATION

3.1 Role of Flutter

Flutter is used to develop a cross-platform mobile application that allows users or caretakers to:

- Configure medicine schedules
- Assign time slots to specific drawer IDs
- Update or delete existing schedules

3.2 Bluetooth communication

The Flutter app connects to the ESP32 using Bluetooth (BLE or Classic Bluetooth). This approach ensures:

- Operation without internet connectivity
- Secure short-range communication
- Low power consumption

3.3 rest api consumption using dio

Although communication is over Bluetooth, the ESP32 exposes REST-style HTTP APIs locally. Flutter uses the Dio package to consume these APIs.
Example responsibilities include:

- Sending schedule data (drawer ID, hour, minute)
- Requesting device status
- Manually triggering drawer opening (optional)

Dio provides structured request handling, JSON serialization, timeout management, and error handling, making it suitable for robust mobile–IoT communication.

4. ESP32-BASED MEDICINE BOX

4.1 ESP32 as iot Controller

The ESP32 microcontroller is selected due to its:

- Built-in Bluetooth and Wi-Fi
- Sufficient processing power
- Low cost and low energy consumption

In this system, ESP32 acts as a standalone IoT controller that does not depend on external servers for normal operation.

4.2 REST API Implementation on ESP32

The ESP32 hosts a lightweight HTTP server and exposes REST APIs such as:

- POST /schedule – Store medicine schedule
- GET /schedules – Retrieve stored schedules
- POST /open Drawer – Open a specific drawer manually

These APIs follow REST principles and exchange data in JSON format.

4.3 schedule persistence

Received schedules are stored locally on the ESP32 using **Little FS or Preferences storage**. This ensures:

- Schedule persistence across power cycles
- Autonomous execution even when the Flutter app is disconnected

5. SCHEDULING AND EXECUTION LOGIC

The ESP32 maintains an internal clock using either:

- Network Time Protocol (NTP), or
- An external RTC module (for offline reliability)

A scheduler routine continuously compares the current time with stored schedules. When a match is detected:

- The corresponding drawer ID is identified
- The motor control function is triggered
- The drawer opens automatically

Each schedule is executed only once per day, with a daily reset mechanism implemented at midnight.

6. DRAWER OPENING MECHANISM

Each medicine drawer is equipped with:

- A servo motor or solenoid latch
- A mechanical latch and spring-based sliding mechanism When triggered by the ESP32:
  1. The servo rotates to unlock the latch
  2. The drawer slides open using spring force or gravity
  3. After a predefined delay, the latch returns to the locked position

Only the drawer with the specified ID is activated, ensuring precise medicine dispensing.

7. ADVANTAGES OF THE PROPOSED SYSTEM

- **Offline Operation:** No continuous internet or backend server required
- **Autonomous Execution:** Schedules run even if the Flutter app is disconnected
- **Low Cost:** Minimal hardware and no cloud infrastructure
- **Scalable:** Can support multiple drawers and schedules
- **User-Friendly:** Simple mobile interface using Flutter

8. LIMITATIONS AND FUTURE ENHANCEMENTS

limitations

- Limited to short-range Bluetooth communication
- No centralized cloud monitoring in current design

Future work

- Integration with cloud backend for remote monitoring
- Caretaker notification for missed doses

- Sensor-based confirmation of pill removal
- Battery-powered portable version

9. CONCLUSION

This research paper presents a Bluetooth-based smart medicine box using Flutter and ESP32 that ensures reliable, autonomous, and accurate medicine dispensing. By leveraging REST APIs on the ESP32 and consuming them through Dio in Flutter, the system achieves a clean separation between user interaction and device control. The offline-first design makes the solution practical for real-world healthcare scenarios, especially for elderly and home-care patients.

10. REFERENCES

[1] Espressif Systems, ESP32 Technical Reference Manual
[2] Google, Flutter Documentation
[3] Arduino, ESP32 Arduino Core
[4] Bluetooth SIG, Bluetooth Low Energy Specification
[5] IEEE, IoT Applications in Healthcare