Project: ***Smart Medicine Box***

Submitted To: ***Sir Nasir Mehmood***

Submitted By: ***Laiba Taseen 23-NTU-CS-1258***

Rabia Naseer 23-NTU-CS-1275

Malaika 23-NTU-CS-1291

Course Title: ***Embedded IOT Systems***

Corse Code: ***CSE-3079***

National Textile University Faisalabad

Problem Statement

Medication problems are common, especially for older people and patients with long-term diseases. People often forget to take their pills, take the wrong ones, or store them in bad conditions like heat or moisture. This can harm their health. Caregivers also cannot always check if medicines are taken on time.

Objectives

- ✓ ***Automated Scheduling:*** *To design a system that automatically unlocks specific medicine compartments at pre-set times.(Medicine schedules are currently hardcoded using relative system uptime (millis)).*
- ✓ ***Multi-Modal Alerting:*** *To provide audible (Buzzer), visual (LED/OLED), and digital (Mobile App) reminders.*
- ✓ ***Verified Compliance:*** *To use IR sensors to detect if the patient has actually reached into the box to take the medicine.*
- ✓ ***Remote Monitoring:*** *To enable caregivers to track pill intake and environmental storage conditions via the Blynk IoT platform.*
- ✓ ***Missed Dose Notifications:*** *To log and alert the caregiver if a dose is not taken within a specific timeframe.*

System Architecture

The system follows a three-layer architecture:

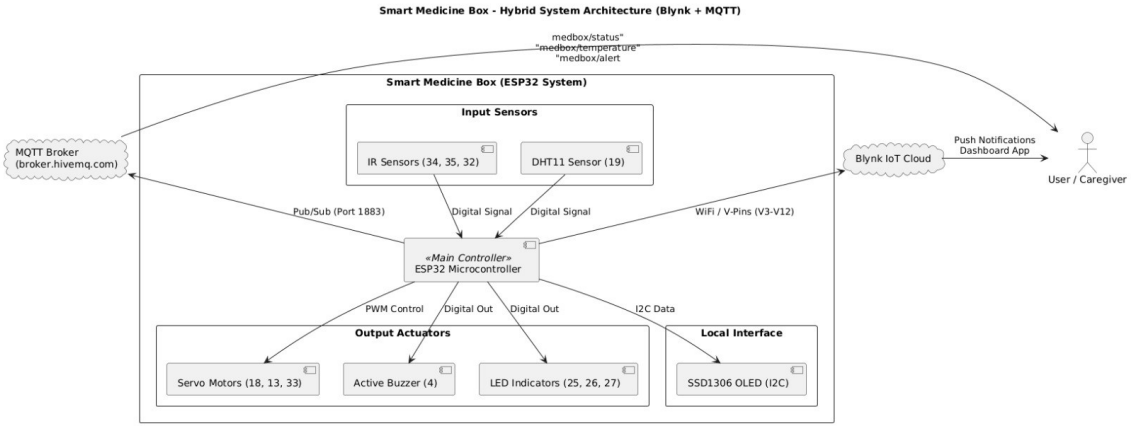
i. ***Input Layer:*** *DHT11 (Climate), IR Sensors (Medicine detection), Blynk App (Control) and Node-Red(MQTT data visualization).*

ii. ***Processing Layer:*** *The ESP32 acts as a dual-client, communicating via the Blynk Protocol and the MQTT Protocol.*

iii. ***Output Layer:*** *Servos (Lid opening), Buzzer/LEDs (Alerts), OLED Display (Status),*

   a. ***Mobile:*** *Blynk App for remote control and notifications.*
   b. ***Desktop:*** *Node-RED Dashboard for advanced MQTT data visualization.*

Block Diagram



Hardware and Software Description

Hardware:

- ***ESP32:*** *The brain with built-in Wi-Fi.*
- ***Servos (x3):*** *Opens specific compartments for different doses.*
- ***IR Sensors (x3):*** *Detects if the patient has physically taken the medicine.*
- ***DHT11 Sensor:*** *Measures storage temperature and humidity.*
- ***OLED Display:*** *Shows local status and sensor readings.*
- ***Buzzer & LEDs:*** *Visual and audible alarms.*

Software:

- ***Blynk IoT:*** *Mobile app interface for remote alerts and monitoring.*
- ***Arduino IDE:*** *Used for writing and uploading the C++ code.*
- ***Libraries:*** *BlynkSimpleEsp32, ESP32Servo, DHT, Adafruit_SSD1306, and PubSubClient (for MQTT communication).*

- ***Protocols:*** *Dual-protocol support using Blynk (Websockets) and MQTT for high reliability and real-time data broadcasting.*

Methodology

i. ***Dual-Initialization:*** *The ESP32 connects to Wi-Fi and simultaneously initializes the Blynk service and the HiveMQ MQTT broker connection.*

ii. ***Telemetry Publishing:*** *Every second, the system reads sensor data (T/H/IR) and publishes it to both the Blynk Dashboard and specific MQTT topics (e.g., medbox/temperature).MQTT publishes the data to Node-RED for desktop monitoring.*

iii. ***Alerting & Verification:*** *When a dose is due, the system triggers the servo, buzzer, and LED. It then publishes a Dispense command to Node-RED via MQTT and logs the event on Blynk.*

iv. ***Confirmation:*** *If the IR sensor detects movement, the system publishes a Medicine Taken status to the broker and resets. If ignored for 60 seconds, it broadcasts a Missed Dose alert to both platforms.*

MQTT Communication

To ensure the system can be monitored by third-party tools (like MQTT Explorer) or other smart home systems, the following data structure is implemented:

MQTT Topic	*Data Content*	*Purpose*
medbox/status	ESP32 Connected / Medicine Taken	General connectivity and activity log.
medbox/temperature	Numeric value (16.7)	Real-time storage temperature tracking.
medbox/humidity	Numeric value (65.5)	Real-time storage humidity tracking.

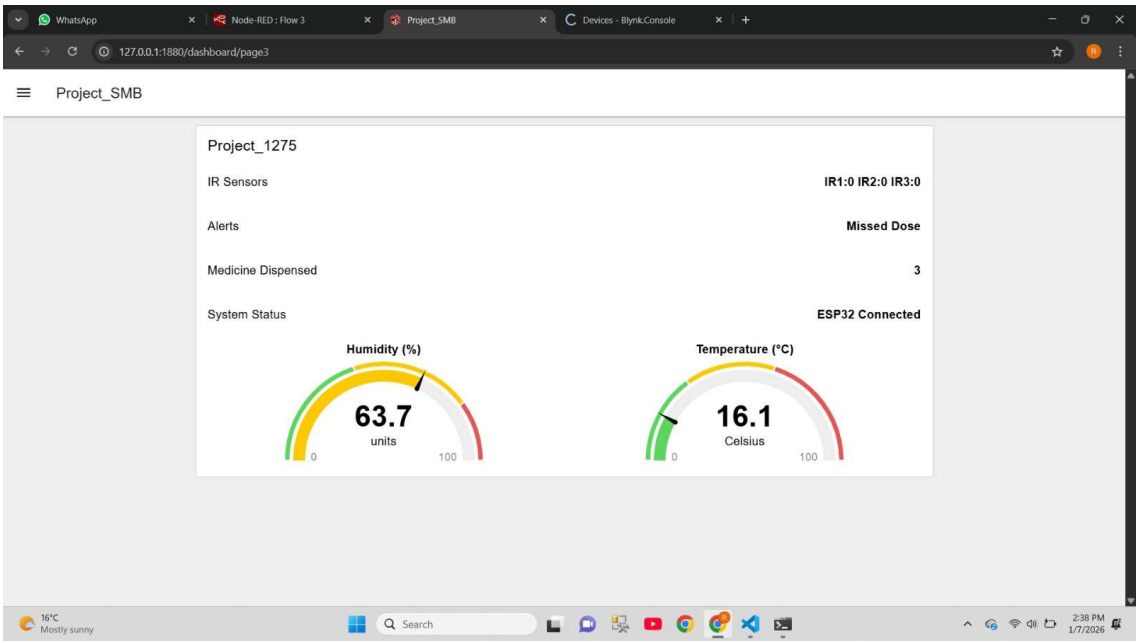
medbox/ir	IR1:1 IR2:0 IR3:0	Raw sensor data for movement detection.
medbox/alert	Missed Dose	Emergency notification for caregivers.
medbox/dispense	1 / 2 / 3	Indicates which compartment is dispensing

Node-RED & MQTT Integration

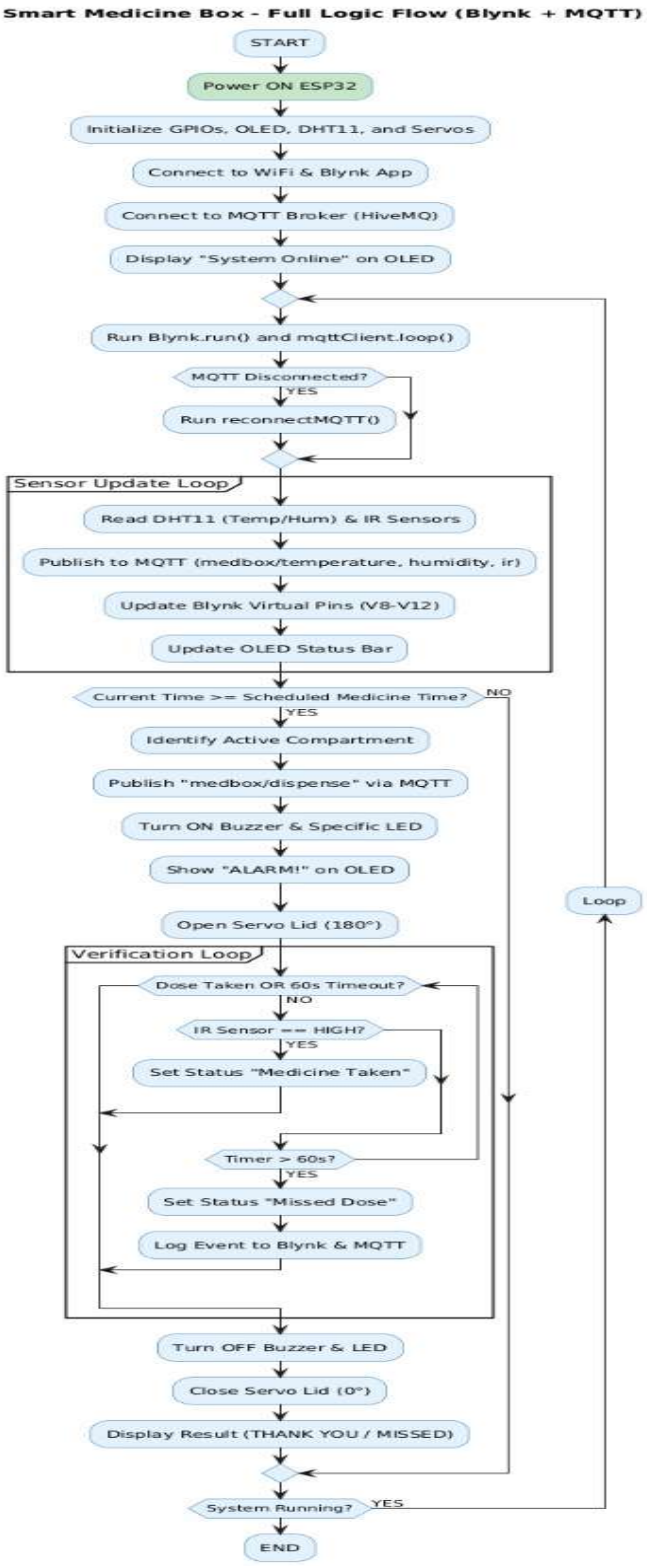
*To provide a more robust monitoring solution, **Node-RED** is used as the central MQTT management platform.*

- ***MQTT Broker:** Data is sent to the broker.hivemq.com server.*
- ***Node-RED Logic:** The MQTT-In nodes subscribe to topics like medbox/temperature, medbox/humidity, and medbox/status.*

i. ***Dashboard Nodes:** Data is piped into Gauges, Charts, and Text boxes to create a real-time health monitoring station.*



Flow chart


Smart Medicine Box - Full Logic Flow (Blynk + MQTT)
START
Power ON ESP32
Initialize GPIOs, OLED, DHT11, and Servos
Connect to WiFi & Blynk App
Connect to MQTT Broker (HiveMQ)
Display "System Online" on OLED
Run Blynk.run() and mqttClient.loop()
MQTT Disconnected?
YES
Run reconnectMQTT()
Sensor Update Loop
Read DHT11 (Temp/Hum) & IR Sensors
Publish to MQTT (medbox/temperature, humidity, ir)
Update Blynk Virtual Pins (V8-V12)
Update OLED Status Bar
Current Time >= Scheduled Medicine Time?
NO
YES
Identify Active Compartment
Publish "medbox/dispense" via MQTT
Turn ON Buzzer & Specific LED
Show "ALARM!" on OLED
Open Servo Lid (180°)
Verification Loop
Dose Taken OR 60s Timeout?
NO
IR Sensor == HIGH?
YES
Set Status "Medicine Taken"
Timer > 60s?
YES
Set Status "Missed Dose"
Log Event to Blynk & MQTT
Turn OFF Buzzer & LED
Close Servo Lid (0°)
Display Result (THANK YOU / MISSED)
System Running?
YES
Loop
END

Screenshots of Output and Dashboards

- ***OLED Screen:*** *Displays ALARM! or "T:25C H:50%".*
- ***Blynk App:*** *Show the Gauges for Temp/Humidity and the Virtual LEDs for IR status.*
- ***Serial Monitor:*** *Show the Wi-Fi connection logs and Box Open messages.*

Results, Conclusion, and Future Scope

Results:

The project successfully dispenses medicine at three different intervals. The Wi-Fi integration allows for real-time monitoring, and the IR sensors provide a reliable way to track patient compliance.

Conclusion:

This IoT device reduces the risk of missed medications. It provides peace of mind for caregivers through automated logs and remote notifications.

Future Scope:

- ***Cloud Scheduling:*** *Allow users to change medicine times via the app (currently hardcoded).*
- ***Voice Integration:*** *Add Alexa or Google Assistant reminders.*
- ***Pill Counting:*** *Add load cells to weigh the medicine and ensure the correct number of pills are taken.*
- ***Database Integration:*** *Use Node-RED to save all medication history into a SQL or NoSQL database for long-term medical reporting.*