

Project Report
On
Smart Medical Dispenser



Submitted
In partial fulfilment
For the award of the Degree of

PG-Diploma in Embedded Systems and Design
(PG-DESD)

C-DAC, ACTS (Pune)

Guided By:

Mr Shripad V Deshpande

Submitted By:

Kampe Naresh (250240130016)

Vaishnavi Sandip Kasabekar (250240130038)

Vikas Gupta (250240130039)

Shinde Priyanka Sunil (250240130042)

Centre for Development of Advanced Computing (C-DAC), ACTS

(Pune- 411008)

Acknowledgement

We would like to express our sincere gratitude to everyone who contributed to the successful completion of our project, “Smart Medical Dispenser.”

First and foremost, we are extremely thankful to our project guide, Dr. Shripad Deshpande, for their valuable guidance, encouragement, and constant support throughout the project. Their expertise and constructive feedback played a crucial role in shaping the direction and outcome of our work. We extend our heartfelt thanks to the faculty and staff of the PG-DESD CDAC ACTS Pune, for providing the infrastructure and resources necessary for carrying out this project. Their technical support and encouragement were instrumental during the development and testing phases. We are also grateful to our peers and classmates for their feedback, collaboration, and motivating discussions. Their suggestions helped us refine our ideas and improve the functionality of the system. Lastly, we thank our families for their unwavering support and understanding during this journey. Their patience and encouragement inspired us to persevere through challenges and strive for excellence.

This project has been a tremendous learning experience, and we are proud to have developed a system that can contribute positively to health care and patient well-being.

ABSTRACT

In today's fast-paced world, patients, especially the elderly and chronically ill—often forget to take their medications on time or in the correct dosage. This can lead to serious health complications or ineffective treatment. To address this problem, our project proposes the design and development of a Smart Medicine Pill Dispenser that automates medicine reminders and indicates correct medicine with correct dose.

This project addresses the critical issue of medication non-adherence, particularly among the elderly and chronically ill, which can lead to serious health complications and treatment failures. We propose the design and development of a Smart Medicine Pill Dispenser, a microcontroller-based system that automates medication reminders and dispensing to improve patient compliance.

The system is built around an ESP32 microcontroller and features a user-friendly interface comprising a 2.4-inch SPI TFT display and a 4x4 matrix keypad. Users can program medication schedules, including dosage times and compartment assignments, with reminders provided through visual (LEDs) and auditory (buzzer) alerts. The dispenser includes four individual compartments, each equipped with an IR sensor to detect and log pill intake. A Real-Time Clock (RTC) module ensures precise scheduling, while AWS IoT Core integration enables remote monitoring and data logging via Wi-Fi, allowing caregivers to track medication adherence in real-time.

The Smart Pill Dispenser is a low-cost, scalable, and effective solution designed to enhance patient safety, reduce the burden on caregivers, and support independent living. This project contributes to the broader goal of integrating smart technology into healthcare to improve patient outcomes and quality of life.

Table of Contents

S. No	Title	Page No.
	Front Page	I
	Acknowledgement	II
	Abstract	III
	Table of Contents	IV - V
1	Introduction	01-02
1.1	Introduction	01
1.2	Objective and Specifications	02
2	Literature Review	03-04
3	Methodology/ Techniques	05-17
3.1	Overview	05
3.2	System Architecture	06
3.3	Hardware Design	07
3.4	Software Design	14
3.5	Cloud Integration using AWS	14
3.6	Implementation Strategy	14
3.7	Techniques Used	17
4	Implementation	18-21
4.1	Hardware Implementation	18
4.2	Firmware Development (Using Arduino IDE):	21
4.3	AWS Cloud Integration	21
4.4	Testing and Validation	21
5	Results	22-25
5.1	Overview	22
5.2	Functional Verification	22
5.3	System Performance	23
5.4	Test Cases	24
5.5	Medical Box	24

6	Conclusion	26-27
6.1	Conclusion	26
6.2	Future Scope	26
7	References	28
7.1	References	28

Chapter 1

Introduction

1.1 Introduction

In recent years, advancements in embedded systems and the Internet of Things (IoT) have revolutionized healthcare delivery, particularly in areas focused on patient monitoring, medication management, and assistive technologies. One of the most critical challenges in modern healthcare is ensuring medication adherence — the practice of taking medications correctly, at the prescribed times and in the correct dosage. This issue is especially prevalent among the elderly, patients with chronic illnesses, and individuals with memory impairments. Traditional pillboxes and manual reminder methods often fall short, leading to missed or incorrect doses that can significantly impact a patient's health, recovery, or disease management. The Smart Medical Pill Dispenser project aims to solve this problem by automating and digitalizing the pill intake process using embedded hardware, sensors, and cloud connectivity. This system is built around the ESP32 microcontroller, a low-cost, Wi-Fi-enabled microprocessor that serves as the central control unit. A 2.4-inch SPI TFT display provides visual feedback and reminders to the user, while a 4x4 matrix keypad allows the user or caregiver to configure schedules, select compartments, and set the number of doses. The system includes four physical pill compartments labeled A to D, each equipped with an infrared (IR) sensor to detect whether the pill has been taken. LEDs and a buzzer are used to alert the user at the scheduled time, and a Real-Time Clock (RTC) module ensures accurate scheduling regardless of power resets or outages. What sets this system apart is its integration with AWS (Amazon Web Services) via Wi-Fi, allowing the ESP32 to communicate with cloud services in real time. When a pill is taken or missed, this information is sent to AWS IoT Core and stored in a DynamoDB database. Additionally, if a scheduled dose is missed, the system triggers an alert using AWS SNS (Simple Notification Service), which can notify caregivers via SMS or email. This cloud connectivity provides remote visibility into the patient's medication behavior and enables caregivers or family members to monitor adherence from anywhere.

This project blends electronics, software, and cloud computing to create a comprehensive health management solution that enhances safety, independence, and convenience for patients. It is scalable, affordable, and designed with flexibility in mind — supporting additional compartments, advanced features like automatic dispensing, or integration with mobile apps in future versions.

In summary, the Smart Medical Pill Dispenser represents a practical and impactful application of IoT technology in the healthcare sector, aiming to improve medication adherence, reduce the burden on caregivers, and ultimately contribute to better patient outcomes.

1.2 Objective

The primary objective of this project is to design and develop a Smart Medical Pill Box that automates the process of medication reminders and intake monitoring, while also enabling remote supervision through cloud connectivity. This system aims to enhance medication adherence and support independent living, especially for elderly patients and individuals with chronic conditions.

The specifications of the project are:

1. To design a medical box using ESP32 that can notify users when it is time to take their medication.
2. To incorporate a 2.4-inch TFT SPI display for real-time user interaction, allowing the user to view schedules, number of dose , and medication prompts clearly.
3. To implement a 4x4 matrix keypad that enables patients or caregivers to easily configure medicine schedules, dosage times, and pill counts directly on the device.
4. To integrate IR sensors in each of the four pill compartments (A, B, C, and D) to detect whether the medicine has been taken or missed.
5. To include LED indicators and a buzzer to visually and audibly alert the user at the scheduled medication times.
6. To use a Real-Time Clock (RTC) module to ensure accurate timekeeping and scheduling, even in the event of a power reset.
7. To establish wireless communication between the device and AWS IoT Core for real-time data transmission.

Chapter 2

LITERATURE REVIEW

The increasing demand for smart healthcare solutions has led to the development of various systems and technologies aimed at improving patient care and medication adherence. A key focus area has been the automation of medicine reminders and dispensers, particularly for the elderly and individuals with chronic diseases. Several research efforts and commercial systems have addressed this challenge using microcontrollers, sensors, real-time monitoring, and cloud connectivity. This literature review outlines significant contributions in the field and how this project builds upon and improves existing solutions.

1. Traditional Pill Dispensers and Manual Systems

Conventional pillboxes are widely used to organize medications by day or time, but they rely heavily on the user's memory and discipline. These systems are prone to human error, leading to skipped doses or double dosing, particularly among elderly patients. Studies have shown that lack of consistent adherence can result in treatment failure or health deterioration (World Health Organization, 2003).

2. Microcontroller-Based Pill Reminder Systems

Early implementations of digital pill dispensers used simple microcontrollers such as Arduino or PIC controllers to activate buzzers or lights at scheduled times. These systems were effective for local reminders but lacked the ability to track usage or provide remote access. For instance, research by Saravanan et al. (2016) proposed a basic GSM-based pill reminder using timers and alarms, but it lacked real-time verification of pill intake.

3. IoT-Enabled Dispensers with Cloud Integration

Recent developments in IoT and cloud computing have enabled more advanced medication management systems. Espressif's ESP32 microcontroller, with built-in Wi-Fi and low-power operation, has become a popular platform for IoT health applications. Projects such as Smart Medicine Box by Kale et al. (2018) implemented a Wi-Fi-enabled device that alerts users via mobile applications but lacked on-device configurability or pill detection sensors.

4. Pill Intake Verification Using Sensors

IR (Infrared) sensors and load cells have been introduced to detect whether pills have been physically

removed from compartments. A system by Ahmed et al. (2019) used IR sensors to check for the presence of a hand near the pill compartment, improving accuracy in monitoring adherence. However, many such systems lacked proper user interfaces or secure cloud integration for data logging and caregiver alerts.

5. Cloud Services for Health Monitoring

Cloud platforms such as AWS, Google Firebase, and Azure have made it possible to collect and analyze patient health data in real-time. Using AWS IoT Core, DynamoDB, and SNS, developers can build scalable systems for storing pill logs and sending alerts. Systems integrating these services have shown better results in monitoring medication adherence, especially when caregivers receive notifications about missed doses.

6. Gaps and Opportunities

Despite numerous advancements, existing systems often suffer from limitations such as complexity in setup, lack of user-friendly input interfaces, unreliable sensor integration, or absence of robust cloud connectivity. Many require users to pre-program schedules via a computer or app, which may not be accessible to all users.

Chapter 3

Methodology and Techniques

3.1 Overview

The development of the Smart Medical Pill Dispenser is based on the integration of embedded hardware, real-time sensing, user interfaces, and cloud communication. This chapter outlines the step-by-step methodology used to implement the system, highlighting both the hardware and software components, and the techniques applied to ensure reliability, accuracy, and user-friendliness.

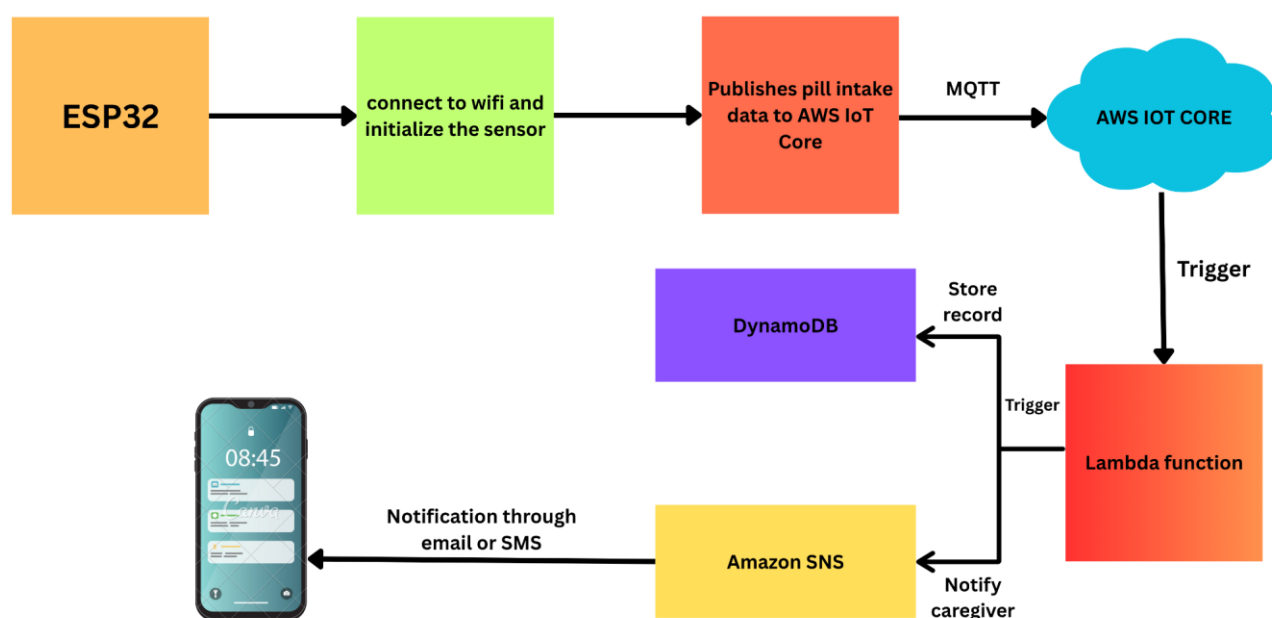


Figure 3.1: Overview of Smart Medical Box

Figure 3.1 illustrates the end-to-end workflow of a Smart Medical Box using ESP32 and AWS services. The ESP32 microcontroller connects to Wi-Fi and initializes sensors to monitor pill intake. It publishes pill status data via MQTT (Message Queuing Telemetry Transport) to AWS IoT Core, which then triggers a Lambda function. The Lambda function processes the data and performs two actions: it stores the pill intake log into Amazon DynamoDB, and if the medicine was missed, it uses Amazon SNS to send real-time notifications (via SMS or email) to the caregiver. This setup ensures secure, cloud-based

medication tracking and timely alerts for missed doses.

3.2 System Architecture

The overall system is composed of four main subsystems:

- Hardware Control Unit (ESP32-based)
- User Interface (TFT Display and Keypad)
- Sensing and Actuation (IR sensors, RTC, buzzer, LEDs)
- Cloud Communication and Monitoring (AWS IoT Core, DynamoDB, SNS)

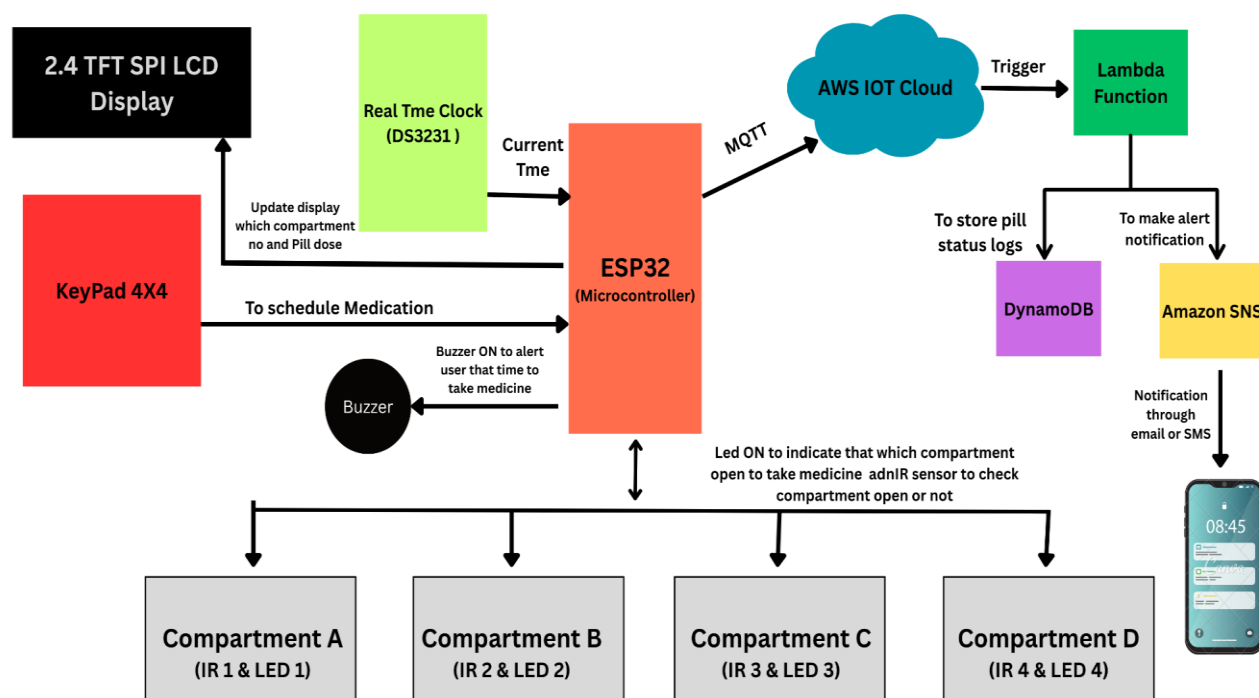


Figure 3.2: System Architecture

The Hardware Control Unit, based on the ESP32 microcontroller, acts as the brain of the system. It orchestrates all operations, including time tracking, signal processing, user inputs, and cloud communications. The ESP32 receives time updates from the Real-Time Clock (DS3231), manages the state of each compartment, and decides when to activate alerts. It communicates with the AWS IoT Core over Wi-Fi using MQTT protocol to publish pill intake data and receive scheduled updates, ensuring

seamless interaction between the local device and the cloud environment.

The User Interface comprises a 2.4-inch TFT SPI display and a 4x4 matrix keypad. The display shows real-time information such as the current time, scheduled medicine alerts, and the status of each compartment. The keypad allows the user (or caregiver) to interact with the system by scheduling medication times, selecting compartments (A–D), and entering the number of doses. This interface is essential for configuring the system and ensuring the patient can operate it independently without needing a mobile app or computer.

The Sensing and Actuation subsystem includes IR sensors, LED indicators, a buzzer, and the RTC. IR sensors detect whether the pill compartment was accessed, confirming if the dose was taken. LEDs light up the appropriate compartment at the scheduled time, and the buzzer provides an audible alert.

The Cloud Communication and Monitoring subsystem, powered by AWS IoT Core, Lambda functions, DynamoDB, and Amazon SNS, ensures that every pill intake or missed dose is logged in the cloud. SNS alerts are sent to caregivers via SMS or email for any missed medication, enabling remote health monitoring and timely intervention. Together, these subsystems create a robust, automated, and cloud-connected pill dispensing solution.

3.3 Hardware System Design

The hardware system includes the following major components:

ESP32-S3 Microcontroller:

Espressif Systems created the potent microcontroller and Wi-Fi module known as the ESP32. It is an improved version of the ESP8266 with more features and functionalities. The ESP32's adaptability, low power consumption, and simplicity of usage make it a popular choice for Internet of Things applications.

The dual-core CPU of the ESP32, which enables effective multitasking, is one of its primary characteristics. For applications that need both processing power and connection, this makes it perfect. Furthermore, the ESP32 has integrated Bluetooth and Wi-Fi, allowing for easy wireless connectivity.

The ESP32's extensive peripheral set, which includes GPIO pins, SPI, I2C, UART, and ADC interfaces,

is another noteworthy feature. As a result, interacting with a variety of sensors, actuators, and other external devices. Along with support for many sleep modes and power-saving measures, the ESP32 is also incredibly programmable. Because of this, it can be used in battery-powered applications where energy economy is essential.

The Arduino IDE, ESP-IDF (Espressif IoT programming Framework), or other programming platforms can be used to program the ESP32. With compatibility for both C and C++, the ESP32 is programmed by a broad spectrum of developers

All things considered, the ESP32 is a strong and adaptable microcontroller that works well with a wide range of Internet of Things applications. Developers wishing to construct connected products and systems frequently choose it because of its low power consumption, connection choices, and computational capability.

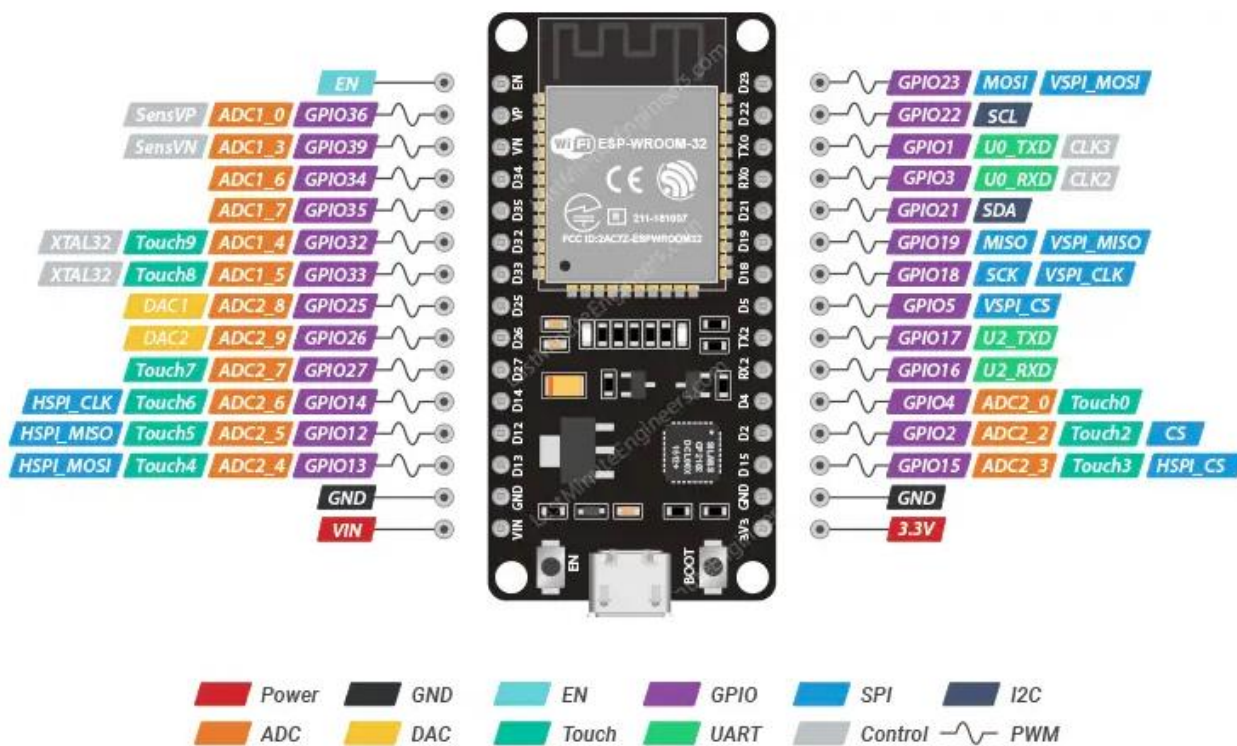


Figure 3.3: ESP32-S Microcontroller

ESP32-S Microcontroller Services are given below:-

- **Central Controller:** Serves as the main controller for all components of the smart pill dispenser

system.

- **Peripheral Management:** Interfaces with the TFT display, 4x4 keypad, IR sensors, LEDs, RTC, and buzzer.
- **Wi-Fi Communication:** Utilizes built-in Wi-Fi to securely connect to AWS IoT Core and publish MQTT messages.
- **Dual-Core Processing:** Enables efficient multitasking—simultaneously handles sensor input, display updates, and cloud messaging.
- **Cloud Integration:** Communicates with cloud services such as AWS DynamoDB and SNS for real-time data logging and alerts.
- **Low Power Operation:** Supports various low-power sleep modes, making it suitable for energy-efficient applications.
- **Flexible Programming:** Compatible with Arduino IDE and ESP-IDF using C/C++, which allows wide accessibility to developers.

2.4-inch TFT SPI Display:

A 2.4-inch TFT SPI display is a compact, color screen module popular in DIY electronics and embedded systems. TFT stands for Thin-Film Transistor, which is the technology used to create the vibrant, sharp colors on the LCD screen. What makes this display particularly easy to use is its communication method: it uses a Serial Peripheral Interface (SPI). This means it only requires a few wires to connect to a microcontroller, such as an Arduino or Raspberry Pi, which saves valuable pins and simplifies the wiring process. Most of these displays have a resolution of 240x320 pixels and are controlled by a dedicated driver chip, with the ILI9341 being a very common example.

To use one of these displays, you connect the SPI pins (like CS, SCK, and MOSI) to your microcontroller and then use a software library specifically designed for the display's driver chip. Libraries like those from Adafruit make it straightforward to draw graphics, text, and images on the screen. The combination of its small size, color capabilities, and simple SPI interface makes it an excellent choice for a wide range of projects, from displaying sensor data and creating simple user interfaces to building small games or digital gauges.

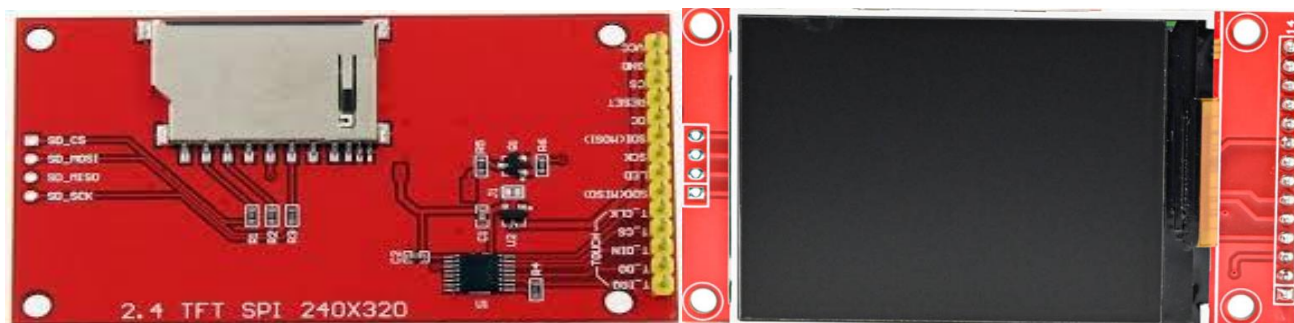


Figure 3.4: 2.4-inch TFT SPI Display

The 2.4-inch TFT SPI Display serves as the primary visual interface for the Smart Pill Dispenser system. It provides users with real-time information such as current date and time, scheduled pill reminders, the active pill compartment, and system feedback (e.g., successful pill intake, missed doses, or error messages). This enhances usability by ensuring that the patient or caregiver can clearly understand system instructions and respond promptly.

The display is connected to the ESP32 microcontroller via the SPI (Serial Peripheral Interface) protocol, which ensures fast and efficient communication. The key signal lines used include MOSI (Master Out Slave In), SCK (Serial Clock), CS (Chip Select), DC (Data/Command), and RST (Reset). These connections allow the ESP32 to control the display efficiently and update content dynamically in response to system events.

4x4 Matrix Keypad:

The keypad consists of 16 buttons arranged in 4 rows and 4 columns. It is connected to the ESP32 microcontroller via 8 digital GPIO pins—4 for rows and 4 for columns. The ESP32 scans the keypad by sequentially pulling each row low and checking which column line reads low, thereby detecting the specific key pressed. This method is both memory and CPU-efficient, making it ideal for embedded applications.

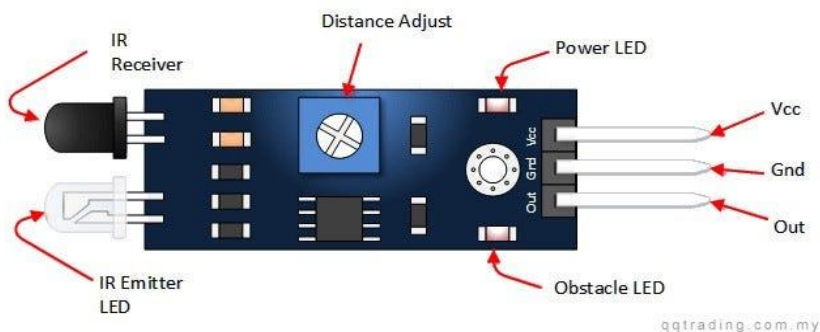


Figure 3.5: 4x4 Matrix Keypad

The 4x4 Matrix Keypad is a critical input device in the Smart Pill Dispenser system, enabling users to interact with the system easily. It allows the patient or caregiver to set medication schedules, select pill compartments (A, B, C, D), and enter the number of pills to refill or confirm actions. Its tactile interface simplifies user navigation through the system menu shown on the TFT display.

IR Sensors:

An Infrared (IR) sensor is an electronic device that detects and measures infrared radiation. All objects with a temperature above absolute zero (0 Kelvin) emit some form of thermal radiation in the infrared spectrum, which is invisible to the human eye. IR sensors are designed to sense this radiation, allowing them to detect the presence of objects, motion, heat, and even measure temperature without physical contact. The fundamental principle is based on the interaction between an infrared source and a receiver, or simply by the receiver detecting ambient IR radiation.



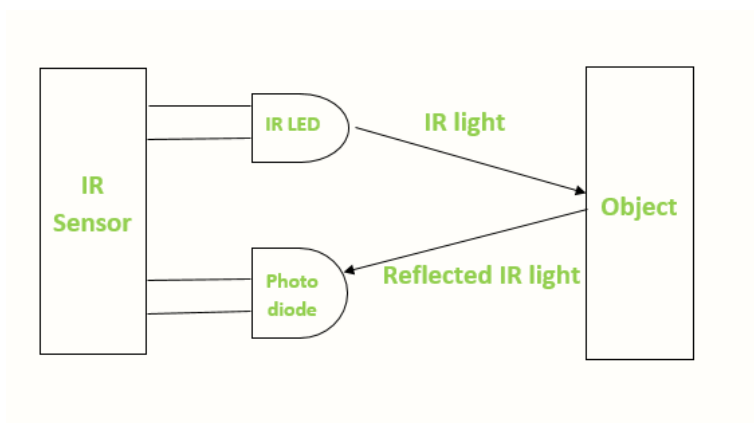


Figure 3.6: IR Sensor

The Smart Pill Dispenser system uses Infrared (IR) Sensors, with one sensor placed for each pill compartment (A–D). These sensors play a crucial role in ensuring medication adherence by detecting whether the user has accessed the pill compartment at the scheduled time.

Each IR sensor works by emitting infrared light and detecting its reflection. When a hand or object is placed near the pill compartment to take the medicine, the IR sensor detects the change in reflected IR light, confirming that the compartment was accessed. This data is used by the ESP32 to determine whether the medication was "taken" or "missed", which is then logged and optionally sent to the cloud for caregiver monitoring.

The sensors are connected to the ESP32 through digital GPIO pins, allowing real-time monitoring of compartment interaction.

LED Indicators and Buzzer:

The Smart Pill Dispenser includes LED indicators and a buzzer to provide clear, real-time alerts during scheduled medication times. Each pill compartment (A, B, C, and D) is equipped with a dedicated LED. When it's time to take a pill from a specific compartment, the corresponding LED turns on, guiding the user to the correct section of the dispenser. This visual cue ensures that the patient does not confuse compartments or miss a scheduled dose.



Figure 3.7: Led and Buzzer

In addition to the LEDs, a buzzer is used to provide an audible alert whenever a scheduled medication time arrives. The buzzer serves as an immediate attention signal, especially for elderly users or those who might not notice visual prompts alone. The buzzer and LEDs are both controlled by the ESP32 via its digital GPIO pins, allowing the system to trigger alerts in sync with the schedule set by the user. These alert mechanisms work together to enhance usability and reliability of the dispenser.

Real-Time Clock (RTC) – DS3231:

The module communicates with the ESP32 via the I2C protocol, using dedicated SDA and SCL lines. This interface allows the ESP32 to fetch the current time and date whenever needed, particularly to check if it's time to trigger an alert or mark a pill as taken or missed. The use of the DS3231 RTC guarantees reliable and consistent scheduling, which is critical for timely medication reminders in healthcare applications.

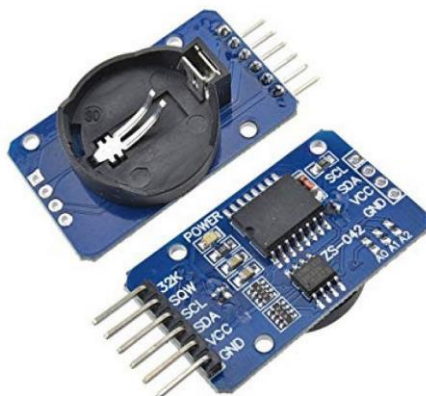


Figure 3.8: Real-Time Clock (RTC) – DS3231

The DS3231 is a highly accurate Real-Time Clock (RTC) module used in the Smart Pill Dispenser to maintain precise timekeeping. Even when the ESP32 microcontroller is powered off, reset, or restarted, the DS3231 continues to track the current date and time using its onboard battery backup. This ensures that medication schedules remain accurate and are not lost due to power interruptions.

3.4 Software Design

The software stack includes:

Embedded C/C++ for ESP32 (Arduino IDE):

- Controls all hardware components.
- Includes logic for comparing real-time with scheduled medication time.

3.5 Cloud Integration using AWS

AWS IoT Core:

- Serves as the MQTT broker for communication between ESP32 and AWS cloud.

AWS Lambda:

- Triggered by incoming MQTT messages.
- Parses payload and stores data in DynamoDB.
- Send an SMS or email alert if a pill is missed using SNS.

DynamoDB:

- Stores pill intake logs in a structured format for future retrieval.

SNS (Simple Notification Service):

- Sends notifications (SMS/email) to caregivers when a dose is missed.

3.6 Implementation Strategy

❖ Flow Diagram:

A process is represented graphically by a flowchart. This kind of diagram illustrates a procedure or workflow. Another definition of a flowchart is a diagrammatic description of an algorithm, or a

methodical process for completing a job. The stages are represented by different types of boxes in the flowchart, and their sequence is indicated by arrows linking the boxes

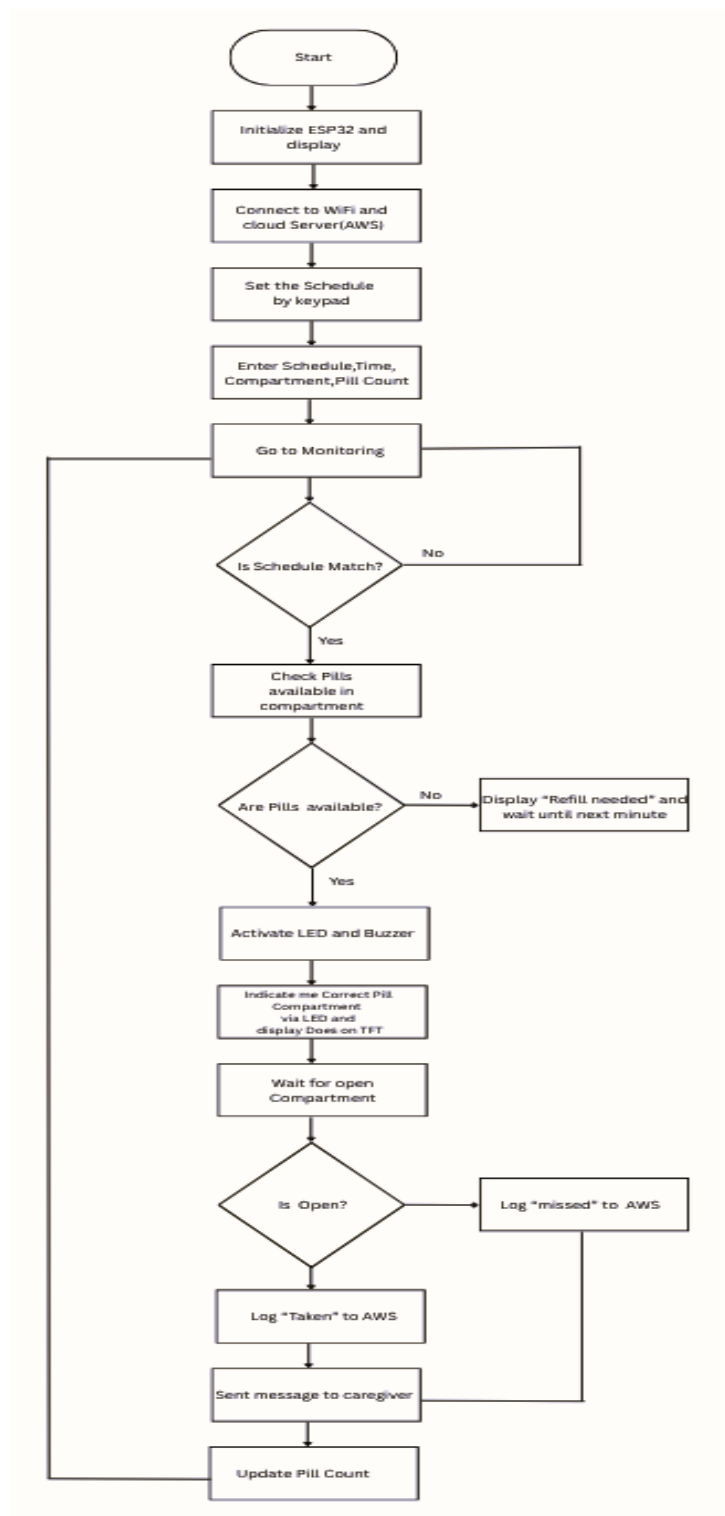


Figure 3.9: Flow Diagram

1. Initialization & Setup Phase

- The system starts by initializing the ESP32 microcontroller and the TFT display.
- It then connects to Wi-Fi and establishes a connection with the cloud server (AWS IoT Core) to enable remote monitoring and communication.
- The user sets the medication schedule using the 4x4 matrix keypad by entering:
 - Schedule time
 - Compartment (A to D)
 - Pill count
- Once scheduling is completed, the system enters monitoring mode.

2. Monitoring & Detection Phase

- The ESP32 continuously monitors the real-time clock (RTC) to check whether the current time matches the scheduled medication time.
- If the schedule does not match, the system waits and rechecks periodically.
- If the schedule matches, it proceeds to check whether pills are available in the selected compartment.

3. Alert & Logging Phase

- If pills are not available, the system:
 - Displays "Refill needed" on the TFT screen.
 - Waits until the next scheduled time.
- If pills are available, the system:
 - Activates LED and buzzer to notify the user.
 - Indicates the correct compartment on the TFT screen (e.g., Compartment A).
- It then waits for the compartment lid to open, monitored via an IR sensor.

4. Logging & Notification Phase

- If the compartment is not opened within a set time, it logs the event as “missed” on AWS and ends the cycle.
- If the user opens the compartment, it logs the action as “taken” to AWS.

- Regardless of the outcome, the system sends a message to the caregiver using Amazon SNS (email or SMS).
- Finally, the pill count is updated in the system, and it returns to monitoring for the next scheduled dose.

❖ Data Flow Diagram

A data-flow diagram is a visual aid used to illustrate how data moves through a system or process. It illustrates information inputs, yields, capacity focuses, and the paths between each goal using described pictures such as squares, circles, and bolts together with brief text markings. It shows how the other components interface and is shown by arrows that are usually identified with a brief data name.

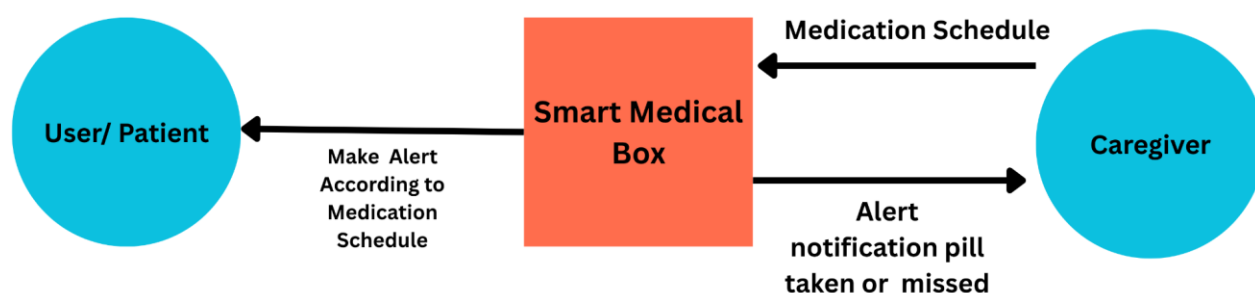


Figure 3.10: Level 0 Data Flow Diagram

3.7 Techniques Used

- Polling and Interrupts: Used for keypad scanning and IR sensor reading.
- I2C and SPI Communication: Efficient bus communication with RTC and TFT.
- MQTT Protocol: Lightweight IoT protocol for cloud data transmission.
- JSON Encoding: Used for structured message formatting.
- Debouncing: Handled keypad noise to ensure accurate key detection.

Chapter 4

Implementation

4.1 Hardware Implementation:

- Connect ESP32-S3 microcontroller to all peripheral devices (TFT display, keypad, IR sensors, buzzer, LEDs, RTC).
- Integrated 2.4" SPI TFT display for real-time UI using SPI protocol.
- Wired 4x4 matrix keypad to GPIO pins for schedule and input settings.
- Installed IR sensors in four compartments (A–D) to detect pill intake.
- Connected RTC (DS3231) via I2C for accurate time management.
- Added LEDs (per compartment) and a buzzer for alert notifications.

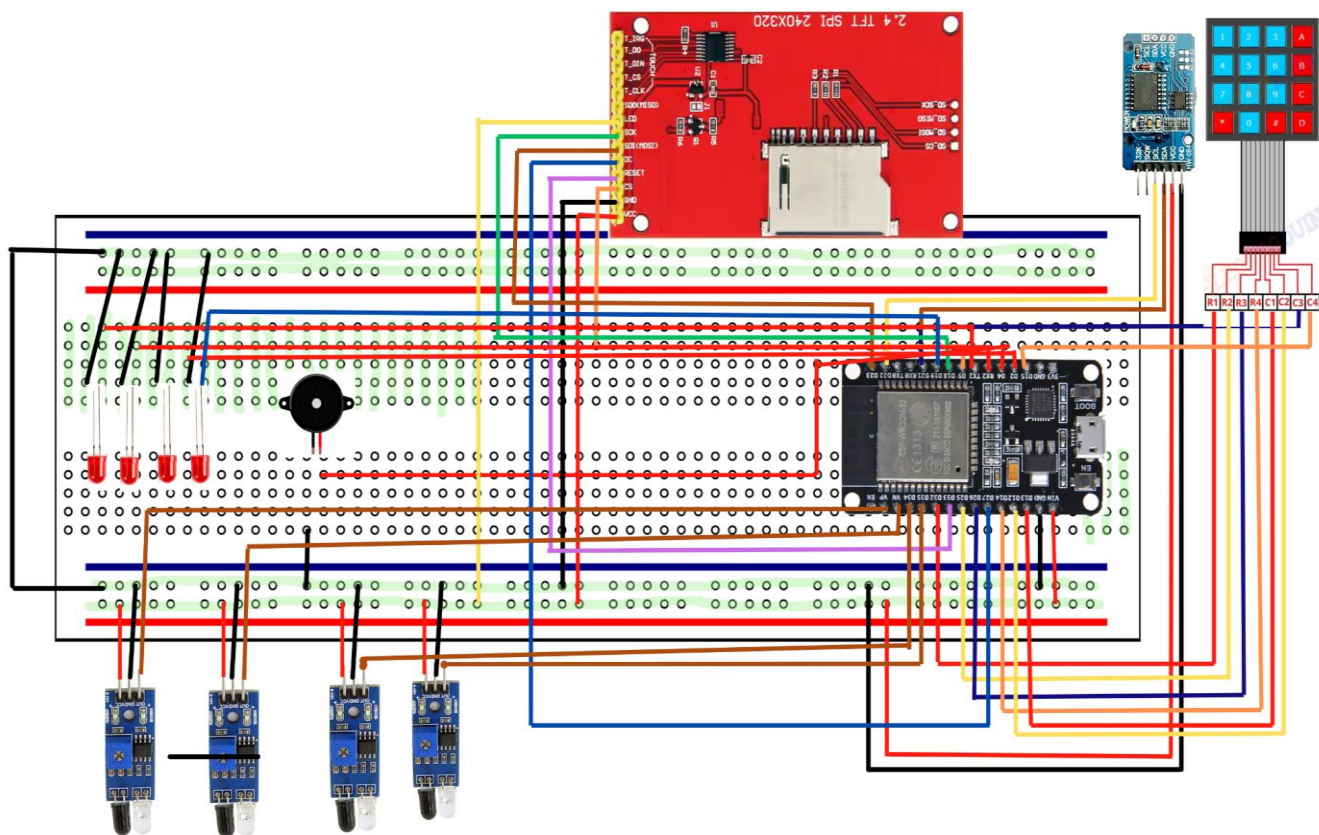


Figure 4.1: Circuit Diagram

The entire circuit is built on a breadboard using a common power (VCC) and ground (GND) . The ESP32's VIN and GND pins power these rails, which in turn supply power to all connected components.

ESP32 Pinout Summary:-

The following table summarizes the connections from the ESP32 to each component:

Component	Function	ESP32 GPIO Pin
TFT Display	Chip Select (CS)	GPIO 5
	Data/Command (DC)	GPIO 27
	Reset (RST)	GPIO 33
	MOSI (SPI Data)	GPIO 23
	SCK (SPI Clock)	GPIO 18
	LED (Backlight)	VCC (Common Power)
Keypad 4x4	Row 0	GPIO 32
	Row 1	GPIO 25
	Row 2	GPIO 26

	Row 3	GPIO 14
	Column 0	GPIO 13
	Column 1	GPIO 12
	Column 2	GPIO 21
	Column 3	GPIO 15
Buzzer	Signal Pin	GPIO 4
LED Indicators	Compartment A	GPIO 16
	Compartment B	GPIO 17
	Compartment C	GPIO 2
	Compartment D	GPIO 19
IR Sensors	Compartment A	GPIO 34 (input only)
	Compartment B	GPIO 35 (input only)
	Compartment C	GPIO 36 (input only)
	Compartment D	GPIO 39 (input only)

RTC (DS3231)	SDA (I2C Data)	GPIO 21
	SCL (I2C Clock)	GPIO 22

4.2 Firmware Development (Using Arduino IDE):

- Wrote initialization and communication routines for all peripherals.
- Programmed keypad logic to accept schedule times and dose input.
- Implemented IR sensor monitoring and compartment status tracking.
- Developed real-time alert logic using RTC data comparison.
- Created TFT UI to display pill prompts, system time, and alerts.

4.3 AWS Cloud Integration:

- Configured AWS IoT Core for MQTT communication with ESP32.
- Created secure device credentials and uploaded to ESP32 flash.
- Designed Lambda function to parse incoming pill data.
- Stored pill activity logs in AWS DynamoDB.
- Used AWS SNS to send SMS/email alerts if a pill is missed.

4.4 Testing and Validation:

- Verified individual components with sample test codes.
- Simulated daily pill schedules and monitored alerts.
- Validated cloud logs and confirmed remote alert delivery.
- Ensured IR sensors accurately detected pill removal.
- Conducted system-wide integration tests under real-world conditions.

Chapter 5

Results

5.1 Overview

The Smart Medical Pill Dispenser system was designed, developed, and tested successfully to meet the intended objectives of improving medication adherence, enabling real-time monitoring, and integrating cloud-based alerts. This chapter presents the observed outcomes, performance metrics, and user response during the testing and demonstration phases.

5.2 Functional Verification

All key components were tested in an integrated environment, and the system performed as expected under controlled test conditions:

Display Functionality

- The 2.4" TFT SPI screen displayed time, schedules, and alert messages correctly.
- The interface was responsive and readable, even under various lighting conditions.

Keypad Input

- The 4x4 matrix keypad accurately captured user inputs for time, compartment selection, and dosage.
- Input debounce and error handling were functional and user-friendly.

Real-Time Clock (RTC)

- The DS3231 module maintained precise time, including after ESP32 resets or power cycles.
- Time comparison and schedule triggering matched defined user inputs.

IR Sensors

- The IR sensors accurately detected pill removal in each compartment (A–D).
- Minimal false positives or missed detections were observed during repeated tests.

Alerts

- LEDs illuminated correctly based on compartment-specific reminders.
- The buzzer produced clear audio alerts at scheduled times.
- Alerts deactivated after pill intake or timeout.

AWS Cloud Connectivity

- ESP32 reliably connected to AWS IoT Core via Wi-Fi.
- MQTT messages containing pill activity were successfully published.
- Data logged into DynamoDB was consistent and queryable.

Remote Notifications

- AWS SNS alerts (SMS or email) triggered within seconds when a pill was missed.
- Caregivers received timely notifications with relevant pill and time data.

5.3 System Performance

- System boot time: < 3 seconds
- Time to send MQTT message to AWS: < 1 second
- Alert trigger response time: Instant (< 0.5 sec after schedule match)
- IR detection accuracy: ~98% in controlled tests
- Cloud log success rate: 100% for valid pill events

5.4 Test Cases

Test Case	Input	Expected Output	Result
Pill alert at scheduled time	Schedule for 08:00 AM	LED and buzzer ON	Pass
IR detection after pill taken	Hand near compartment A	Mark pill as "taken"	Pass
Missed pill (no IR detection)	Timeout after alert	Mark pill as "missed"	Pass
MQTT payload format	JSON published to AWS	Received in correct structure	Pass
SNS alert after missed dose	No pill taken	SMS/email sent to caregiver	Pass

5.5 Smart Medical Box



Figure 5.1: Side View of Smart Medical Box



Figure 5.2: Top View of Smart Medical Box

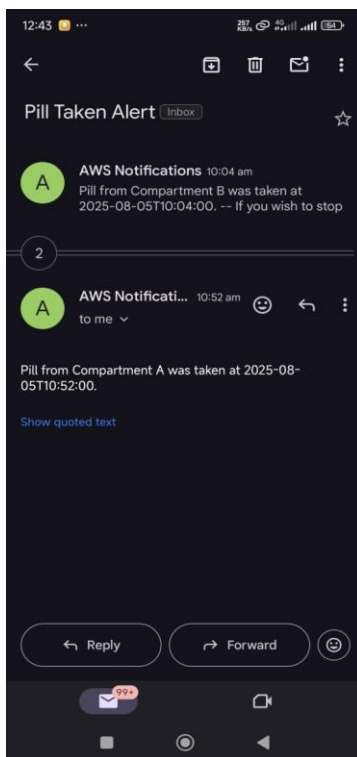


Figure 5.3: Screen Short of Mail notification

Chapter 6

Conclusion

6.1 Conclusion

The Smart Medical Pill Dispenser project successfully automates medication reminders and monitors pill intake using ESP32, IR sensors, a TFT display, and cloud connectivity via AWS. The system alerts users with LEDs and a buzzer, detects pill removal, and notifies caregivers in real-time if a dose is missed.

Key Achievements:

- Accurate scheduling with RTC and user input via keypad.
- Real-time detection of pill intake using IR sensors.
- Cloud integration with AWS IoT, DynamoDB, and SNS for alerts.
- Simple, affordable, and scalable solution for home or clinical use.

Challenges included sensor tuning and AWS setup, but all objectives were met. The project enhances medication adherence and lays a strong foundation for future features like mobile apps or automatic pill dispensing.

6.2 Future Scope

The Smart Pill Dispenser system holds immense potential for future enhancements and real-world applications. In the upcoming versions, integration with a dedicated mobile application can enable remote monitoring, scheduling, and dosage adjustments by caregivers or doctors. This would improve accessibility for elderly and chronically ill patients.

The system can also be extended with voice assistance (e.g., Alexa, Google Assistant) and camera modules for facial recognition to ensure the correct person receives the medication. Furthermore, incorporating machine learning algorithms could enable intelligent analysis of user adherence patterns and help predict potential health risks based on missed dosages.

Lastly, the system can be scaled for use in hospitals, care homes, and pharmacies, where multi-patient support, cloud-based health record integration, and inventory tracking would further enhance its utility and reliability in modern healthcare ecosystems.

Chapter 7

References

- [1] World Health Organization. (2003). Adherence to Long-Term Therapies: Evidence for Action. World Health Organization.
<https://apps.who.int/iris/handle/10665/42682>
- [2] Saravanan, T., Kumar, R., & Priya, M. (2016). GSM Based Pill Reminder for Patient. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 5(2), 196–200.
- [3] Kale, M., Bhoyar, P., & Dhandre, A. (2018). IoT Based Smart Medicine Box. International Journal of Computer Sciences and Engineering, 6(4), 167–172.
- [4] Ahmed, M., Asraf, H. M., & Yasir, M. (2019). IoT Based Pill Reminder and Monitoring System for Smart Health Care. International Conference on Electrical, Computer and Communication Engineering (ECCE), 1–6.
- [5] Espressif Systems. (2021). ESP32 Technical Reference Manual.
https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- [6] Adafruit Industries. (2022). 2.4” TFT LCD Display Guide.
<https://learn.adafruit.com/adafruit-2-4-tft-touch-screen>
- [7] AWS IoT Core Documentation. Amazon Web Services.
<https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html>
- [8] DS3231 RTC Module Datasheet. Maxim Integrated.
<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>
- [9] MQTT Protocol. MQTT Version 3.1.1 Standard.

<http://mqtt.org/>

- [10] Arduino Documentation.
<https://www.arduino.cc/reference/en/>