RESEARCHERID
THOMSON REUTERS
[Javed et al.,11 (5), May 2026]

ISSN:2455-9679
SJIF Impact Factor:6.190

IJRTSM
INTERNATIONAL JOURNAL OF RECENT TECHNOLOGY SCIENCE & MANAGEMENT

"MULTI FUNCTION ALARM CLOCK"

Javed qureshi [1], Bhavika Gajbhiye[2], Palak Marskole [3], Trisha Gavande[4] ,Hariom solanki[5]
[1-5]Sagar Institute of Science, Technology & Research (SISTec-R), Bhopal, Madhya Pradesh, India
jq774009@gmail.com, bhumikapal002@gmail.com,
palakmarskole35@gmail.com, gavandetrisha947@gmail.com

ABSTRACT

Harmony AI is a cognitive agentic workflow system designed to eliminate the friction between conversational messaging and calendar management. By delivering an enterprisegrade intelligent scheduling assistant accessible entirely through WhatsApp, this project directly supports UN Sustainable Development Goal (SDG) 8 (Decent Work and Economic Growth). It empowers micro-entrepreneurs, freelancers, and small businesses to maximize operational productivity by automating administrative overhead without requiring users to switch digital contexts or applications. The core backend architecture integrates the whatsapp-web.js library for real-time message ingestion, a Large Language Model (LLM) served via OpenRouter (utilizing the nvidia/nemotron-3-nano-30b-a3b model) for natural language understanding, and the Google Calendar API for programmatic event management. Furthermore, the project advances SDG 9 (Industry, Innovation, and Infrastructure) through its highly optimized cloud deployment strategy. To overcome the strict memory constraints typical of accessible cloud hosting, the architecture employs a decoupled remote browser strategy via WebSockets and Browserless.io, promoting sustainable, resource-efficient digital infrastructure. User session states and multi-turn conversational histories are securely persisted in a MongoDB Atlas cloud database via a RemoteAuth strategy.

The system's cognitive engine classifies incoming messages into three primary intents: read (schedule inquiry), book (event creation), or cancel (event deletion). For booking requests, a structured extraction pipeline dynamically parses the guest email, event date, start time, duration, and agenda. A conflict detection module queries the Google Calendar free/busy API in real-time; should a scheduling conflict arise, a custom Slot Hunter Algorithm automatically calculates and proposes alternate 30-minute time windows. To ensure data integrity, a two-stage Host Confirmation Pause workflow guarantees that no calendar action is executed without explicit user approval. Deployed entirely on Render cloud infrastructure with Node.js and Express.js serving as the central communication hub, end-to-end testing confirmed the successful, autonomous execution of all core scheduling workflows. The system demonstrates robust natural language comprehension, effectively processing complex intents in both English and Hinglish.

Key Words: *Cognitive Agentic Workflow, WhatsApp Integration, Large Language Model, Google Calendar API, Slot Hunter Algorithm*

ISSN:2455-9679
SJIF Impact Factor:6.190

I. Introduction

The Multi Alert Clock system is an advanced IoT-based embedded system with a dual-microcontroller architecture using ESP32 and Arduino Nano, designed for smart time management with real-time clock display, countdown timer, and audio alerts. Traditional clocks lack remote control and intelligent scheduling, so this project integrates IoT to enable wireless control via a web interface hosted on ESP32. Users can start, pause, resume, and stop timers through a mobile or browser GUI. The Arduino Nano handles countdown logic, RTC module, 7-segment display, and DFPlayer Mini for audio alerts, while UART serial communication ensures fast data transfer between controllers. Operating in clock, countdown, and alarm modes, the RTC ensures accuracy during power interruptions, and multiplexed displays provide clear output with effective audio notifications

II. Review of Literature

The development of IoT-based smart time management systems and AI-driven scheduling assistants has been explored across multiple research domains, with relevant contributions in embedded systems, cloud-based calendar automation, and conversational AI.

IoT-Enabled Smart Clocks and Time Management Systems

Recent projects have demonstrated the integration of ESP32 microcontrollers in smart alarm clocks and time management devices. GitHub repositories such as ESP32-SmartAlarm showcase multi-timezone alarm clocks with MQTT-based synchronization and web-based user interfaces for remote control . Similarly, Cloc—an ESP32-based alarm clock—implements automatic time setting via NTP servers and a fully configurable web server accessible across devices, highlighting the growing demand for wireless control and intelligent scheduling features . The Multi Alert Clock system extends these capabilities by employing a dual-microcontroller architecture (ESP32 and Arduino Nano) to support countdown timers, RTC-based timekeeping, and audio alerts via DFPlayer Mini, thereby addressing the limitations of traditional digital clocks.

Dual-Microcontroller Communication and Synchronization

Research on embedded system synchronization underscores the challenges of coordinating multiple microcontrollers. Arduino community discussions indicate that serial communication (UART) between controllers remains a standard approach, as implemented in this project, while wireless protocols such as ESP-NOW are also gaining traction for distributed IoT deployments . A review on clock synchronization in industrial IoT highlights the difficulty of maintaining unified time perception across resource-constrained devices, emphasizing the importance of reliable serial protocols and RTC modules for accuracy during power interruptions .

III. Problem identification

Conventional alarm clock systems have several limitations. They typically provide only basic time display and simple alarm functionality, which is insufficient for modern applications requiring advanced scheduling and automation.
There is a lack of systems that support remote control, countdown customization, and multi mode operation in a single compact device. In many cases, users need separate devices for timers, alarms, and scheduling, which increases complexity and reduces efficiency.
Additionally, traditional systems rely on manual input buttons and buzzer-based alarms, which are not suitable for noisy environments or remote usage. There is a need for a smart system that integrates real-time clock functionality, countdown timers, and audio alerts with wireless control capability.
This project addresses these issues by developing an IoT-based Multi Alert Clock system using ESP32 and Arduino Nano, enabling wireless control, automation, and enhanced functionality in a single embedded platform.

IV. Solution

The scope of this project extends to educational institutions for scheduling classes and exams, industries for process timing, hospitals for medication reminders, and homes for personal productivity. The system can expand into a full IoT

 ISSN:2455-9679
SJIF Impact Factor:6.190

ecosystem by integrating cloud services, mobile apps, and voice control, making it highly scalable for future automation. The RTC module ensures precise timekeeping during power interruptions using a backup battery. Multiplexed 7-segment LED displays reduce pin usage while maintaining efficiency, and the DFPlayer Mini enables audio playback for effective alerts. Overall, this cost-effective, compact solution offers advanced time management across educational, industrial, and domestic environments.

V. Proposed Methodology

The circuit diagram of the Multi Alert Clock system represents a dual-microcontroller architecture consisting of ESP32 and Arduino Nano. The ESP32 is responsible for WiFi- based communication and web server control, while the Arduino Nano handles real-time processing of timer operations, display control, and alarm generation.
The system integrates multiple hardware modules including RTC module for timekeeping, 7-segment display for visual output, DFPlayer Mini for audio alerts, and transistor driver circuits for display multiplexing. Communication between ESP32 and Arduino Nano is established using UART serial communication.

Main components used in circuit:
1. ESP32 WiFi Module (Web Server + IoT Control Unit)
2. Arduino Nano (Main Processing Controller)
3. RTC Module (DS3231 / RTC307 equivalent – Real Time Clock)
4. 4-digit / 6-digit Multiplexed 7-Segment Display Unit
5. BD136 PNP Transistors (Digit Switching Driver)
6. NPN Transistors (BC547 / similar – Segment Control & Switching)
7. DFPlayer Mini Module (Audio Playback System)
8. Micro SD Card (Stores Audio Files for DFPlayer)
9. Push Buttons (Manual Input / Backup Control Interface)
10. Speaker / Buzzer Output Unit (Alarm Indication)
11. UART Communication Interface (ESP32 ↔ Arduino Nano)
12. 5V Regulated Power Supply Unit

CIRCUIT DIAGRAM-

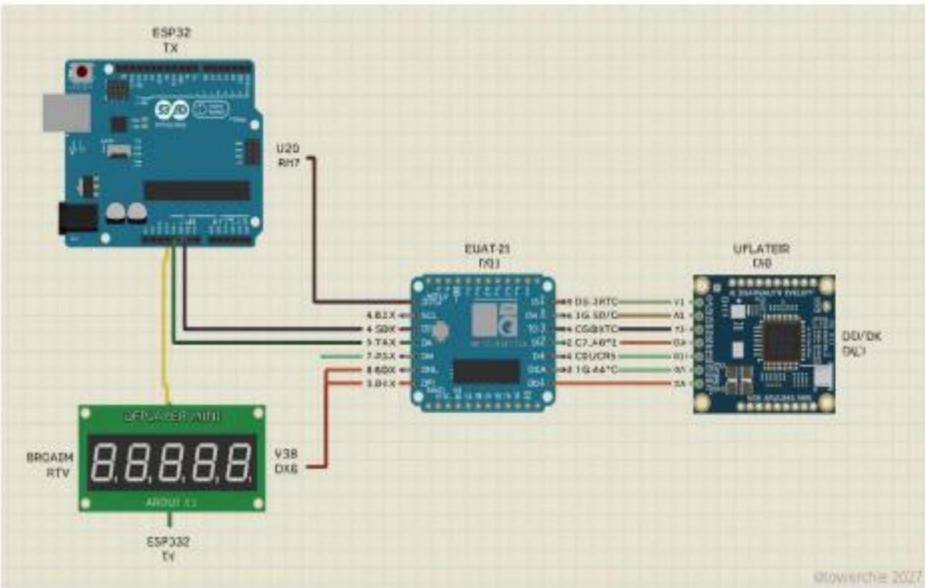

Fig4.1: Circuit Connection Diagram

Working of Circuit Diagram The working of the Multi Alert Clock circuit is based on a dual-microcontroller IoT architecture where ESP32 handles wireless communication and Arduino Nano performs real-time hardware control. The system starts when the ESP32 connects to a WiFi network and hosts a web-based user interface. The user accesses this interface using a mobile phone or browser and inputs timer values or control commands such as start, pause, resume, or stop.
Once a command is given, the ESP32 processes the HTTP request and converts it into a serial UART command. This

ISSN:2455-9679
SJIF Impact Factor:6.190

command is transmitted to the Arduino Nano through UART communication pins. The Arduino Nano receives these commands and executes the required operation, such as starting the countdown timer, pausing execution, resuming operation, or stopping and resetting the system.
During operation, the Arduino Nano continuously reads data from the RTC module to maintain accurate timing. It updates the 7-segment display using multiplexing technique to show either current time or countdown values. When the timer reaches zero, the Arduino activates the DFPlayer Mini module, which plays an audio file through a speaker or buzzer, generating a clear alarm signal.
This working mechanism ensures smooth coordination between wireless control (ESP32) and hardware execution (Arduino Nano), making the system efficient, responsive, and suitable for IoT-based applications.
1. ESP32 WiFi Module The ESP32 is a powerful microcontroller with built-in WiFi and Bluetooth capabilities. In this project, it acts as the web server and IoT communication unit. It hosts a web-based user interface that allows users to control the system remotely using a mobile phone or browser. It receives user commands and sends them to the Arduino Nano using UART communication.
2. Arduino Nano The Arduino Nano is the main processing controller of the system. It executes all realtime operations such as countdown timing, RTC reading, display control, and alarm triggering. It receives commands from ESP32 and controls all connected hardware components accordingly.
3. RTC Module (DS3231 / RTC307 Equivalent) The RTC (Real Time Clock) module is used to maintain accurate time even when the system is powered off. It provides real-time hours, minutes, and seconds data to the 19 Arduino Nano using the I2C communication protocol. It ensures precise timing for clock and scheduling operations.
4. 7-Segment Display Unit (4-digit / 6-digit) The 7-segment display is used for visual representation of time and countdown values. It displays numeric data in a clear format. Since multiple digits are used, multiplexing technique is applied to control all digits using fewer microcontroller pins.
5. BD136 PNP Transistors BD136 transistors are used as digit switching drivers in the display system. They help in controlling which digit is active at a particular time during multiplexing. They also provide required current amplification for proper display brightness. 6. NPN Transistors (BC547 or similar) NPN transistors are used for segment control and switching operations. They help in driving individual segments of the display safely by limiting current and protecting the microcontroller pins.
6. NPN Transistors (BC547 or similar) NPN transistors are used for segment control and switching operations. They help in driving individual segments of the display safely by limiting current and protecting the microcontroller pins.
7. DFPlayer Mini Module The DFPlayer Mini is a compact audio module used for generating sound alerts. It plays preloaded audio files stored in a micro SD card when the timer reaches zero or when an alarm condition occurs. It is controlled by Arduino via serial communication.
8. Micro SD Card The micro SD card is used to store audio files required by the DFPlayer Mini module. These audio files are played during alarm events to provide clear and customizable sound output.
9. Speaker / Buzzer The speaker or buzzer is the final output device that produces audible alerts. It is connected to the DFPlayer Mini and generates sound when the alarm or timer completion event is triggered.
10. 5V Regulated Power Supply The power supply provides a stable 5V DC output required for the operation of the Arduino Nano, RTC module, display system, and audio module. A stable power source is essential for reliable and continuous system performance.

VI. RESULT

The Multi Alert Clock system is an IoT-based embedded project developed using ESP32 and Arduino Nano. It integrates wireless communication, real-time clock functionality, countdown timer operation, and audio-based alert system into a single smart device. The system allows users to control timer operations remotely using a web interface hosted on ESP32, making it more advanced than traditional alarm clocks.
The system provides multiple benefits due to its modern IoT-based architecture. It supports wireless control through WiFi, allowing users to operate the device using a mobile phone or browser from anywhere within the network range. It combines multiple functions such as real-time clock, countdown timer, and alarm system in a single unit, reducing the need for separate devices. The use of RTC module ensures highly accurate timekeeping even during power failure. DFPlayer Mini provides clear audio alerts, improving notification effectiveness compared to basic buzzer systems. The system is also low-cost, easy to implement, and suitable for educational, industrial, and personal productivity applications.

VII. CONCLUSION

The Multi Alert Clock system successfully demonstrates the integration of IoT and embedded system technology. ESP32 handles wireless communication and web-based control, while Arduino Nano performs real-time processing and hardware control. The system improves traditional timing methods by adding remote access, automation, and multi-function capability. It provides accurate timing, reliable performance, and userfriendly operation. Overall, the project is efficient, practical, and

ISSN:2455-9679
SJIF Impact Factor:6.190

suitable for real-world applications in education, industry, hospitals, and home automation.

REFERENCES

- Arduino Official Documentation
  https://www.arduino.cc/en/Guide/HomePage
- ESP32 Technical Documentation (Espressif Systems)
  https://www.espressif.com/en/products/socs/esp32
- DS3231 RTC Module Datasheet
  https://datasheets.maximintegrated.com/en/ds/DS3231.pdf
- DFPlayer Mini MP3 Module User Manual
  https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299
- Embedded Systems Concepts and Applications (Textbook Reference)
- Internet of Things (IoT) Development Tutorials
  https://www.tutorialspoint.com/internet_of_things/index.htm
- Arduino and ESP32 Project Development References (Community Projects)