International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 08 | Aug 2026 www.irjet.net p-ISSN: 2395-0072

Automated College Bell System with Wireless Control

A Microcontroller- and Wi-Fi-Based Approach for Automated Academic Scheduling

Saraswathi R Devadiga[1], Karthik[2], Poornesh[3], Ayush[4]

[1,2]Department of BCA, Trisha Vidya College of Commerce and Management, Katpady, Karnataka, India.

Abstract-*Timely management of academic periods is essential for maintaining discipline and minimizing interruptions in educational institutions. Conventional manual bell operation depends on staff availability and may introduce delays or inconsistencies. This paper presents the design of an Automated College Bell System with Wireless Control using an Arduino-based controller, a 16×2 I2C LCD, tactile switches, an alert actuator, and a wireless communication module. The system allows a timetable to be configured locally and supports wireless commands for manual ringing during events, assemblies, or other exceptional situations. The control software is organized as a finite-state process that initializes the interface, accepts schedule configuration, continuously compares the current time with scheduled periods, and activates the bell for a defined duration when a match occurs. The proposed architecture can be extended with an ESP8266 Wi-Fi module, NTP time synchronization, a relay interface, and mechanical actuation. The project demonstrates a practical and low-cost approach to reducing manual intervention in institutional timekeeping. The available project material describes the intended testing areas but does not provide numerical test measurements; therefore, this paper reports the design and implementation framework without inventing performance statistics.*

Keywords: Automated bell system, Arduino Uno, wireless control, ESP8266, IoT, academic scheduling, automation, microcontroller.

1. INTRODUCTION

Time management is a fundamental requirement in educational institutions because classes, breaks, examinations, assemblies, and other activities are governed by predefined schedules. The project source identifies manual bell operation as a common approach that can result in inconsistencies, delays, and distractions. An automated bell can provide a repeatable mechanism for announcing period transitions while reducing dependence on continuous human intervention.

The proposed system combines microcontroller-based scheduling with wireless control. An Arduino Uno is used as the central processing unit, while a 16×2 LCD with an I2C interface provides a local display and three tactile switches provide configuration input. A piezo buzzer represents the alert mechanism in the prototype; a relay interface can be used when the system is connected to a higher-power AC bell. Wireless connectivity is described using a module such as the ESP8266 Wi-Fi module, enabling remote commands from a smartphone or other user interface.

The main objective is to develop a simple, configurable, and extensible bell-control architecture that can operate according to a timetable while retaining a manual override facility. The paper focuses on the system architecture, hardware interfacing, software logic, wireless operation, implementation considerations, and a practical testing framework.

2. PROBLEM STATEMENT

Manual operation of institutional bells requires a responsible person to monitor the timetable and activate the bell at appropriate times. This introduces the possibility of late or missed bell events and makes timetable changes dependent on human intervention. A suitable automated system should therefore provide scheduled operation, simple timetable configuration, local status display, and a mechanism for authorized manual control when required.

3. OBJECTIVES

- To design a microcontroller-based system that automatically activates a bell at scheduled times.
- To provide a simple local interface for configuring and viewing academic periods.
- To introduce wireless control for manual ringing and remote operation.
- To minimize dependence on continuous manual bell operation.
- To provide a scalable architecture that can be extended to a relay-driven bell and multi-building deployment.
- To define a testing and calibration procedure covering timing, wireless range, latency, and fail-safe operation.

4. SYSTEM ARCHITECTURE

The proposed architecture consists of a processing layer, user-interface layer, communication layer, and actuation layer. The Arduino Uno processes schedule information and input events. The LCD displays the current period and configuration information. Tactile switches are used to navigate the scheduling interface. The wireless module provides communication with a remote user interface. The alert output may drive a piezo buzzer in the prototype or a relay-controlled AC bell in a full-scale installation.

Layer	Component	Function
Processing	Arduino Uno	Executes timing logic, reads inputs, and controls outputs.
Display	16×2 LCD with I2C	Displays current period and configuration menus.
Input	Three tactile switches	Allows local navigation and period configuration.
Wireless	ESP8266 / compatible module	Receives remote commands and supports network connectivity.
Actuation	Piezo buzzer / relay + AC bell	Provides the audible bell indication.
Optional sensing	IR sensor	Can be used for detecting a manual or environmental trigger.
Optional mechanical actuation	Servo motor	Can provide controlled mechanical movement of a bell mechanism.

5. HARDWARE DESIGN AND INTERFACING

5.1 Arduino Uno

The Arduino Uno functions as the central controller. It receives input from the tactile switches and wireless interface, processes the configured timetable, and controls the alert output. The source material describes the Arduino as the 'brain' of the system because it performs the time-sensitive control logic.

5.2 16×2 LCD with I2C Interface

The LCD provides a local visual interface for showing the current period and configuration menus. The I2C interface reduces the number of Arduino data connections required for the display from six to two signal lines, SDA and SCL, thereby simplifying wiring.

5.3 Tactile Switches

Three tactile switches are configured with internal pull-up resistors. They provide a compact user interface through which an operator can enter and navigate the 'Set Periods' configuration mode.

5.4 Alert and Switching Stage

A piezo buzzer is used as the prototype alert/load indicator. For a practical institutional installation, the controller output should interface with an appropriately rated relay module and electrical protection arrangement before driving a high-voltage AC bell. The controller should not directly switch a high-voltage load.

5.5 Wireless Module

The project identifies ESP8266 Wi-Fi as a suitable wireless bridge. It can communicate with the Arduino through serial TX/RX communication and can receive commands from a smartphone interface. The original material also mentions HC-05 Bluetooth as an alternative example; the final hardware choice should be fixed before deployment.

5.6 IR Sensor and Servo Motor

The source material also describes an IR sensor and servo motor as part of an extended architecture. The IR sensor can detect an object or trigger condition, while the servo can provide controlled angular movement, typically through PWM. These components are useful when the audible bell is implemented as a physical mechanical mechanism rather than only an electronic buzzer.

6. SOFTWARE DESIGN AND CONTROL ALGORITHM

The software is organized as a finite-state control process. During initialization, the controller configures I2C communication and the required input/output pins. The system then enters a configuration state in which the operator can define the daily periods. After configuration, the main loop continuously checks the current time against the stored schedule.

When the current time matches a scheduled period, the controller activates the bell output for a predefined duration and then returns the output to its inactive state. This prevents the bell from remaining continuously active. The project also proposes the use of real-time clock data or network time synchronization for accurate timekeeping.

7. WIRELESS CONTROL AND IOT INTEGRATION

Wireless control extends the system beyond fixed local operation. A wireless module connected to the controller can accept commands such as a manual ring request. This is useful when a bell must be activated outside the normal timetable, for example during an assembly or emergency.

For a Wi-Fi implementation, the ESP8266 can provide network connectivity and can be integrated with a cloud or mobile interface. The source material proposes NTP synchronization so that the controller can obtain real-time information over the institutional network. Such synchronization can reduce dependence on manually maintained clock settings.

8. IMPLEMENTATION METHODOLOGY

Implementation can be divided into three layers: hardware integration, software deployment, and testing/calibration. During hardware integration, the Arduino, wireless module, display, switches, and alert stage are connected and powered according to their electrical requirements. If a servo or Wi-Fi radio causes significant current demand, the power arrangement must be designed to avoid controller instability.

During software deployment, the C++ firmware is uploaded to the Arduino. The timetable is stored as configurable period data, and the main control loop evaluates the schedule continuously. For network-enabled operation, NTP synchronization may be used to obtain current time. Manual override commands should be processed separately from scheduled events so that authorized users can ring the bell when required.

9. OPERATIONAL WORKFLOW

1. Power on the system and initialize the Arduino, LCD, input switches, wireless interface, and alert output.
2. Enter the period-configuration mode and define the required bell schedule.
3. Store the configured schedule in the controller's available memory.
4. Display the current period or configuration status on the LCD.
5. Continuously compare the current time with each scheduled bell time.
6. When a scheduled time is reached, activate the bell output for the configured duration.
7. Return the output to the inactive state after the ringing interval.
8. Accept an authorized wireless manual-ring command when an unscheduled bell is required.
9. Use manual override or local control as a fallback if network connectivity is unavailable.

10. TESTING AND EVALUATION FRAMEWORK

The project source identifies three major testing areas: latency testing, range validation, and fail-safe verification. Because the supplied project material does not include measured numerical results, this paper does not assign performance values that were not experimentally reported.

Test	Purpose	Method	Expected Observation
Schedule accuracy	Verify scheduled ringing	Compare actual activation time with configured time	Bell activates at the intended schedule.
Wireless latency	Measure command-to-ring delay	Send repeated manual-ring commands and record response time	Consistent and acceptable response delay.
Wireless range	Validate connectivity	Test operation at increasing distances from the access point/module	Stable communication within the intended campus area.
Fail-safe operation	Verify fallback behavior	Disconnect network and test local/manual operation	System remains controllable without network dependency.
Output duration	Check ringing interval	Measure active time of buzzer/relay output	Output returns to inactive state after the configured duration.
Power stability	Check controller reliability	Observe operation under simultaneous wireless/actuator activity	No resets or unstable behavior during normal operation.

10. RESULTS AND DISCUSSION

The project demonstrates a feasible architecture for automating institutional bell operation using widely available embedded-system components. The design directly addresses the identified limitation of manual bell operation by transferring timetable execution to a programmable controller. The wireless interface additionally provides flexibility for exceptional events and schedule changes.

The available source states that the system improves time management and reduces manual errors and delays; however, no numerical before-and-after measurements, sample size, timing-error dataset, or statistical analysis are provided. Therefore, the appropriate research interpretation is that the prototype establishes the mechanism and intended benefits, while quantitative performance validation remains part of future experimental work.

The combination of local configuration and wireless control is particularly useful because it avoids making the system completely dependent on a remote network. A local interface can support basic operation, while wireless communication can

provide convenience and authorized override functions. For deployment at scale, electrical isolation, secure wireless access, reliable time synchronization, and a documented manual fallback procedure are important engineering considerations.

12. ADVANTAGES

- Reduces the need for continuous manual bell operation.
- Provides programmable and repeatable period scheduling.
- Offers local configuration and status display.
- Supports wireless manual control for exceptional events.
- Uses low-cost and widely available microcontroller components.
- Can be extended to relay-driven bells and network-based synchronization.
- Can support future multi-building synchronization through a centralized time source.

13. LIMITATIONS

- The supplied project document does not specify a finalized wireless module; ESP8266 and HC-05 are presented as alternatives.
- The supplied material does not report numerical experimental measurements for latency, range, or timing accuracy.
- The prototype alert mechanism is described as a piezo buzzer, whereas a full-scale installation requires an appropriately isolated and rated relay/AC bell interface.
- Network-based time synchronization depends on Wi-Fi availability and correct NTP configuration.
- Wireless control requires appropriate authentication and access control before institutional deployment.
- The project document describes both buzzer-based operation and an extended IR/servo mechanism; the exact final hardware configuration should be fixed and documented in a production version.

14. FUTURE SCOPE

Future development can include a dedicated mobile application for timetable configuration, real-time status monitoring, and authorized manual control. Cloud-based synchronization can support multiple buildings or campuses from a centralized time source. Additional improvements could include event logging, role-based access control, battery-backed timekeeping, network-failure alerts, and a web dashboard for administrators. A larger experimental study could measure schedule error, wireless latency, connection reliability, and long-term operational stability over several weeks.

15. CONCLUSION

The Automated College Bell System with Wireless Control provides a practical embedded-systems approach to academic time management. The proposed design combines an Arduino controller, LCD-based local interface, tactile switches, scheduled software logic, wireless communication, and an alert actuation stage. By automating scheduled bell events and supporting authorized manual override, the system can reduce dependence on manual operation and provide a more consistent mechanism for period transitions.

The project establishes a useful prototype architecture rather than a fully quantified field study. Its next stage should therefore focus on experimental validation, security, electrical safety, network reliability, and measured performance. With these additions, the system can be developed into a robust and scalable institutional automation solution.

REFERENCES

[1] Arduino CC. “Arduino UNO Official Documentation and Hardware Specifications.” Available online.

[2] Espressif Systems. “ESP8266 Wi-Fi Module Datasheet and AT Instruction Set.” Available online.

[3] Monk, Simon. Programming Arduino: Getting Started with Sketches. McGraw-Hill Education, 2016.

[4] TowerPro. “SG90 Micro Servo Motor Specifications and Pulse Width Modulation Guide.”

AUTHOR INFORMATION

Author: Saraswati
Team Members: Karthik, Poornesh, Ayush