www.ijarp.com ISSN 2456-9992 Page: 01-18

International Journal Advanced Research Publications

Research Article

Volume: 02
Issue: 05

OPERATING A BELL USING ARDUINO BOARD

Shubhangi Balte*[1], Mrunal Gholap[2], Sakshi Wandhekar[3], Dr. S. M. Walke[4]

Department of Electronics and Telecommunications Engineering, Shri Chhatrapati Shivaji Maharaj College of Engineering, Nepti, Ahilyanagar, Maharashtra, India.

Article Received: 22 March 2026, Article Revised: 12 April 2026, Published on: 02 May 2026

***Corresponding Author: Shubhangi Balte**

Department of Electronics and Telecommunications Engineering, Shri Chhatrapati Shivaji Maharaj College of Engineering, Nepti, Ahilyanagar, Maharashtra, India.

DOI: https://doi-doi.org/101555/ijarp.7201

ABSTRACT

This project, titled "Operating a Bell using Arduino Board," is designed to automate the control of a bell using an Arduino microcontroller. The main objective is to develop a simple and efficient system that can ring a bell at predefined times or based on user input. In this system, the Arduino board acts as the central controller, processing programming instructions and activating the bell through a relay module or buzzer. This system can be effectively used in schools, colleges, and offices to automate bell operations, thereby reducing manual effort and improving time accuracy. The proposed system is cost-effective, reliable, and easy to implement. It also helps in understanding the fundamental concepts of embedded systems, programming, and automation.

***KEYWORDS*:** Arduino UNO, Real Time Clock (RTC), Relay Module, Automatic Bell System, Arduino programmed, Transistor Switching.

I. INTRODUCTION

In today's world, time management is very important. Efficient time management plays a crucial role in institutional environments. Everything should be performed in time & accurately. Nowadays school or college bells are manually operated. Hence there is a big question of accuracy. Also, it requires manpower and increases cost. Hence here we should use an automatic control system, which saves our manpower and money & also highest accuracy.

In this project, the scope is to design an "Automatic College Bell" & its Implementation on Arduino Uno Board. An Automatic College Bell [1] is a digital circuit that is used for the

purpose of automatic switching of bell as per the given schedule without any human intervention. Generally, wherever we may go, it might be a school or an organization if start or stop of any process is to be conveyed to a large number of people, a bell is used over there which signals the start or stop of any process. So, all these bells are generally operated by the humans directly which is not advisable always as it is not efficient and even accuracy of the time is also being changed. So, in order to avoid this automation-based bell system is to be introduced. [2]. Automation or automatic control, is the use of various control systems for operating equipment such as machinery, processes in factories, boilers and heat-treating ovens, switching on telephone.

Networks, steering and stabilization of ships, aircraft and other applications with minimal or reduced Human intervention. Some processes have been completely automated. The biggest benefit of automation is that it saves Labor; however, it is also used to save energy and materials and to improve quality, accuracy and precision. [3]

II. LITERATURE SURVEY

Automation technologies are increasingly being adopted to simplify tasks and enhance operational efficiency. Among various platforms, the Arduino Uno has emerged as a preferred choice in embedded system development due to its cost-effectiveness and ease of implementation, as introduced by Massimo Banzi.

Earlier bell systems were primarily based on conventional timer circuits, which often lacked precision and adaptability. Such limitations made them less suitable for environments requiring strict time management. In contrast, Arduino-based systems offer a more flexible and programmable approach, enabling accurate scheduling and control of operations, as highlighted by Simon Monk.

The inclusion of Real-Time Clock (RTC) modules further strengthens the system by providing precise and continuous time tracking, even during power interruptions. This combination ensures dependable performance in time-sensitive applications. The Arduino official resources also emphasize the significance of RTC modules in maintaining real-time accuracy.

Several academic studies, including works like *"Automatic School Bell System Using Microcontroller"*, have demonstrated that automated bell systems improve reliability while minimizing the need for manual intervention. Therefore, this project is designed to develop a highly efficient, reliable, and economical automatic bell/buzzer system using Arduino and RTC technology.

III. Problem Statement

In many educational institutions, bells are operated manually to indicate the start and end of classes. This method is inefficient and prone to human errors, such as delays or missed timings. It also requires continuous human involvement, which increases operational effort.

Additionally, manual systems lack flexibility and cannot be easily modified for special events, examinations, or schedule changes. Therefore, there is a need for an automated, reliable, and accurate bell system that can operate based on a predefined schedule without human intervention.

IV. Methodology

The development of the Arduino-based automatic bell system follows a structured approach. Initially, the system design and required components such as Arduino Uno, RTC module, relay module, and power supply are identified.

The hardware components are then connected according to the circuit design. The RTC module provides accurate time data, while the Arduino Uno processes this data and controls the bell operation.

Programming is carried out using Arduino IDE, where the bell schedule is defined. The system continuously compares real-time data with the programmed schedule and activates the bell accordingly. Finally, the system is tested and optimized to ensure proper functionality and accuracy.

A. Block Diagram

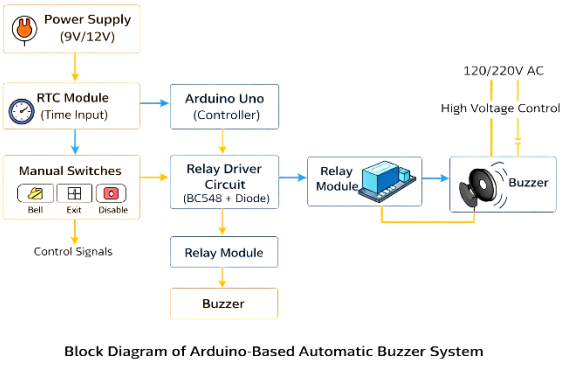

Fig. 1: System Block Diagram of Automatic Bell System.

The block diagram of the **Arduino-based Automatic Buzzer System** illustrates the systematic flow of signals and control in the project. The system is powered by a **9V/12V**

power supply, which provides the necessary voltage to the Arduino UNO and other components. The **RTC (Real Time Clock) module** continuously provides accurate time data to the Arduino, enabling it to ring the buzzer according to the pre-set schedule. **Manual switches** are included to allow users to enable, disable, or trigger the buzzer manually whenever needed. The **Arduino UNO** acts as the main controller, processing inputs from the RTC module and manual switches, and sending control signals to the **relay driver circuit**, which uses a **BC548 transistor and diode** to safely amplify the signal for the relay module. The **relay module** then switches the high-voltage AC circuit, controlling the **buzzer or bell**, which produces sound according to the programmed schedule or user commands. This setup ensures **accurate, safe, and automated control** of the bell system while maintaining flexibility for manual operation. –

B. Circuit Diagram

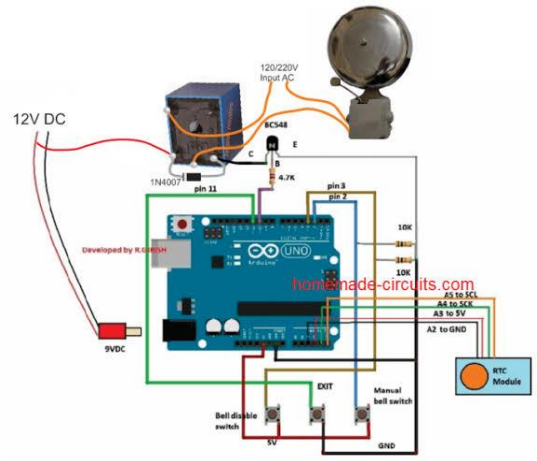

Fig. 2: Circuit Diagram of Automatic Bell System.

The circuit consists of Arduino Uno as the main controller connected with an RTC module for real-time clock input. The RTC module provides accurate time to the Arduino through communication pins.

A relay driver circuit using a BC548 transistor and a 1N4007 diode is connected to the Arduino. The transistor amplifies the signal, and the diode protects the circuit from reverse voltage. The relay module is used to control the high voltage bell/buzzer.

Push buttons are connected for manual control such as bell ON/OFF, exit, and disable functions. The entire system is powered using a 9V/12V DC supply, ensuring proper operation of all components.

C. Pin Configuration Table

Component	Arduino Pin	Connection Type	Description
RTC Module	A4, A5	I2C	SDA → A4, SCL → A5
Relay Module	D8	Digital Output	Controls the bell
Buzzer / Bell	D9	Digital Output	Output device
Transistor BC548	D7	Digital Output	Drives relay
Resistors	Inline	Series	Current limiting
Diode IN4007	Across Relay Coil	Protection	Prevents back EMF

V. HARDWARE REQUIREMENTS

- Arduino Uno
- RTC Module
- Relay Module
- BC548 Transistor
- 1N4007 Diode
- Resistors (10K)
- Resistors (4.7K)
- Adapter 12V
- Buzzer / Electric Bell
- Push Buttons (Switches)

A. *Arduino Uno*

Fig. 3: Arduino uno.

In this system, Arduino Uno acts as a central processing unit that controls all operations used in this project. It is a microcontroller board based on the ATmega328P, which is responsible for controlling all the operations of the system.

The Arduino Uno reads data from input devices like the RTC module and processes it according to the programmed instructions. Based on this, it sends signals to output components such as the relay module to operate the bell.

It has 14 digital input/output pins and 6 analog input pins, which allow easy connection with different components. The board can be powered using a USB cable or an external power supply.

In this Arduino-based bell system, the Arduino Uno acts as the brain of the system. It continuously checks the current time from the RTC module and automatically activates the relay to ring the bell at the correct time.

The Arduino Uno is widely used because it is easy to program, cost-effective, and suitable for beginners as well as advanced projects.

B. *RTC Module*

Fig. 4: RTC Module.

The **RTC Module** (Real Time Clock) is an important component used to keep track of current date and time. It helps the system to operate based on real-time conditions.

The RTC module continues to run even when the main power is off because it has a small backup battery. This ensures that the correct time is maintained without interruption.

In this project, the RTC module is used to store and provide accurate time to the Arduino. Based on this time, the Arduino automatically controls the bell ringing schedule. This makes the system fully automatic and reduces the need for manual operation.

The RTC module communicates with the Arduino using I2C protocol, which requires only two pins (SDA and SCL), making it easy to connect and use.

Overall, the RTC module improves the accuracy and reliability of the Arduino-based bell system by ensuring that the bell rings at the correct time.

C. *Relay Module*

Fig. 5: Relay Module.

The **Relay Module** is an important component used to control high-voltage devices using low-voltage signals. It acts as a switch that allows the Arduino to safely operate electrical devices like a bell.

A relay module works on the principle of an electromagnetic switch. When a small current is applied from the Arduino, the relay gets activated and either completes or breaks the circuit connected to the bell.

The relay module has three main terminals:

- **COM (Common)**
- **NO (Normally Open)**
- **NC (Normally Closed)**

In this project, the relay module is used to turn the bell ON and OFF at specific times. The Arduino sends a signal to the relay through the transistor, and the relay controls the high-power circuit of the bell safely.

The relay module is preferred because it provides electrical isolation between the Arduino and high-voltage devices, ensuring safety and reliability of the system.

D. *BC548 Transistor*

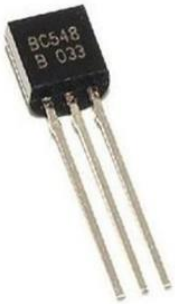

Fig. 6: BC548 Transistor.

The **BC548 Transistor** is a widely used NPN transistor in electronic circuits. It is mainly used for switching and amplification purposes. In this project, the BC548 transistor plays an important role in controlling the relay module.

It works by using a small current at the base terminal to control a larger current flowing between the collector and emitter. This makes it suitable for use as a switch in low-power applications. The transistor has three terminals: Collector, Base, and Emitter.

The BC548 is preferred because it is low cost, easily available, and reliable. It also provides good current gain, which helps in efficient circuit operation.

In the Arduino-based bell system, the transistor is used to amplify the signal from the Arduino and safely drive the relay, ensuring proper functioning of the bell system.

E. ***1N4007 Diode***

Fig. 7: IN4007 Diode.

The **1N4007 Diode** is a widely used general-purpose rectifier diode that allows current to flow in only one direction, protecting sensitive electronic components from reverse voltage damage. It has a maximum repetitive reverse voltage of 1000V and can safely handle a forward current of 1A, making it suitable for low to medium power circuits. In the Arduino-based bell system, the 1N4007 diode is connected across the relay coil as a flyback diode. This is an important safety measure because when the relay switches off, it generates a back EMF (voltage spike) that can damage the Arduino or the controlling transistor. By providing a path for this spike, the 1N4007 diode ensures that the circuit remains safe and reliable. Its low cost, easy availability, and effectiveness make it an essential component in electronic switching and relay control circuits, contributing to the smooth and stable operation of the system.

F. *Resistors (10K)*

Fig. 8: Resistor 10K.

10kΩ resistor is a passive electronic component that resists the flow of electric current by 10,000 ohms. It is widely used in electronic circuits for current limiting, voltage division, and signal conditioning. The resistor comes in various power ratings, commonly ¼ watt (0.25 W) and ½ watt (0.5 W), which indicates how much energy it can safely dissipate without damage.

Resistors have color bands to indicate their value and tolerance. For a 10kΩ resistor with 5% tolerance, the color code is **Brown-Black-Orange-Gold**.

In microcontroller projects like Arduino, 10kΩ resistors are frequently used as **pull-up or pull-down resistors** for switches and buttons. This prevents the input pins from floating and ensures stable HIGH or LOW readings. They are also used in sensor circuits, LED current limiting, and voltage divider networks.

Key Features:

- Resistance: 10,000 Ω (10kΩ)
- Tolerance: ±5% (gold) or ±1% (brown)
- Power rating: ¼ W or ½ W
- Color code: Brown-Black-Orange-Gold (for 5% tolerance)
- Applications: Pull-up/pull-down resistor, LED current limiter, voltage divider, sensor circuits.

G. *Resistors (4.7K)*

Fig. 9: Resistor 4.7K.

A **4.7 kΩ resistor** is a passive electronic component used to limit or control the flow of electric current in a circuit. Its resistance value is 4700 ohms, which allows it to reduce the current to a safe level for other components. Resistors of this value are commonly used in voltage divider circuits, pull-up and pull-down configurations, and in conjunction with transistors to control base current. They are available in various power ratings, typically 1/4 watt for small electronic circuits, and come in axial or surface-mount packages. The resistor works based on Ohm's law, where the voltage across the resistor is proportional to the current flowing through it. In a project like an Arduino-based bell system, a 4.7 kΩ resistor can be used to protect input pins, control LEDs, or manage transistor switching circuits efficiently.

H. *12V Adapter*

Fig. 10: 12v Adapter.

A **12V adapter** is a power supply device that converts the standard AC mains voltage (usually 110V–240V AC) into a stable 12V DC output. It is commonly used in Arduino-based projects to power the microcontroller and other connected components safely.

Key Features:

- **Input:** 110V–240V AC
- **Output:** 12V DC

- **Current Rating:** Usually 1A or 2A depending on project requirements
- **Connector Type:** Barrel jack compatible with Arduino Uno
- **Function:** Provides a constant voltage supply to ensure proper functioning of all modules and prevents damage from overvoltage

Importance in Arduino College Bell System:

The 12V adapter powers the **Arduino Uno**, **RTC module**, and **relay module**, ensuring that the system operates continuously and reliably. It helps in maintaining consistent voltage, which is essential for accurate timing and smooth operation of the bell system.

I. *Buzzer*

Fig. 11: Buzzer.

A **buzzer** is an electronic device that produces sound when an electric current passes through it. It is commonly used in Arduino-based projects to create alarms or notifications. There are two main types of buzzers: **active buzzers**, which have a built-in oscillator and produce sound when DC voltage is applied, making them easy to use with Arduino, and **passive buzzers**, which require a square wave signal from the microcontroller to produce sound. Buzzers generally operate at a voltage range of 3V to 12V DC and have a current rating of 20–30mA. In the Arduino College Bell System, the buzzer is connected via a relay module and is triggered by the Arduino according to the schedule set in the RTC module, signaling the start or end of classes.

J. *Push Button*

Fig. 12: Push Button.

A **push button** is a simple mechanical switch used to make or break an electrical connection when pressed. In Arduino projects, push buttons are often used as input devices to provide manual control or trigger certain actions. When pressed, the button completes the circuit, sending a signal to the Arduino, which can then perform a programmed task. Push buttons usually operate at low voltage (3V–5V DC) and low current, making them suitable for interfacing directly with microcontrollers. In the Arduino College Bell System, a push button can be used to manually trigger the bell in case of emergencies or for testing purposes.

VI. SOFTWARE REQUIREMENT

A. *ARDUINO IDE*

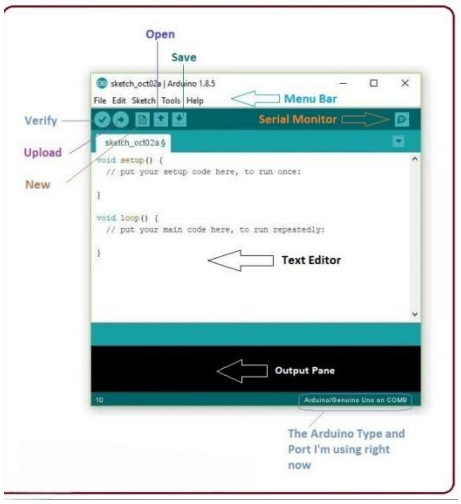

Fig. 13: Arduino IDE.

The Arduino Uno is primarily programmed using the Arduino IDE (Integrated Development Environment), which provides an easy-to-use platform for coding, compiling, and uploading programs (called sketches) to the board via USB. The IDE uses a simplified version of C/C++, making it accessible for beginners while still powerful for advanced projects. Users can control digital and analog pins, interact with sensors and actuators, and manage communication modules using built-in functions. In addition, the Arduino IDE supports libraries, which are pre-written code packages that simplify tasks such as motor control, communication via Bluetooth or Wi-Fi, or working with RTC modules and displays. The IDE also includes a Serial Monitor, allowing users to send and receive real-time data between the Arduino board and the computer, which is useful for debugging and testing. Arduino software is cross-platform, free, and constantly updated.

VII. WORKING PRINCIPLE

Step-by-Step Working Principle

1. Power Supply Conversion:

- The system receives AC mains supply (110V/240V), which is reduced to a lower AC voltage using a step-down transformer.
- This reduced AC is converted into DC using a rectifier diode (1N4007) along with a smoothing capacitor to minimize ripples.
- The resulting DC voltage is used to power the bell (12V DC) and the Arduino Uno (regulated to 9V DC input).

2. Arduino Initialization:

- Once power is applied, the Arduino Uno begins executing the pre-programmed instructions stored in its memory.
- It establishes communication with the RTC module to obtain accurate time data.
- The RTC module continuously maintains real-time information (hours, minutes, seconds) using its internal battery backup.

3. Time Monitoring and Schedule Verification:

- The Arduino continuously retrieves the current time from the RTC module at regular intervals.
- A predefined bell schedule is stored in the program with specific ringing times.
- The system checks whether the current time matches any of the scheduled timings.

4. Relay Activation Mechanism:

- When a match is detected and the system is enabled, the Arduino sends a HIGH signal from digital pin D8.
- This signal is applied to the base of the BC548 transistor through a current-limiting resistor.
- The transistor switches ON, allowing current flow between collector and emitter terminals.
- As a result, the relay coil is energized, causing its internal contacts to close.
- This completes the circuit for the bell, allowing voltage to pass and activate the bell.

5. Manual Bell Operation:

- A push-button switch is connected in parallel with the relay circuit.
- Pressing this switch bypasses the Arduino control and directly activates the bell.
- This feature is useful for manual operation during emergencies or system testing.

6. Bell Ring Duration Control:

- The Arduino maintains the relay in the ON state for a predefined duration (e.g., 5 seconds) as specified in the program.
- After the set time, the Arduino sends a LOW signal to deactivate the relay.
- This stops current flow through the relay, thereby turning OFF the bell.

7. Circuit Protection Mechanism:

- A flyback diode is connected across the relay coil to suppress back EMF generated during switching.
- This prevents potential damage to the transistor and Arduino components.
- Additionally, proper isolation is maintained between low-voltage control circuitry and high-voltage sections to ensure user safety.

8. System Enable/Disable Control:

- An enable switch is interfaced with the Arduino to control system operation.
- When the switch is OFF, the Arduino skips schedule checking and prevents automatic bell activation.
- This feature is beneficial during non-working days or maintenance periods.

VIII. ADVANTAGES

The Arduino-based bell system offers several advantages that make it efficient and reliable for institutions. The system operates automatically based on a predefined schedule, eliminating the need for manual intervention and reducing human effort. By using a Real-Time Clock (RTC), it ensures accurate timing, allowing the bell to ring exactly at the specified time. The system is flexible, as the bell timings can be easily programmed and modified according to changing requirements. It is also cost-effective since it uses Arduino and basic electronic components, making it affordable to implement.

In addition, the system is energy efficient, consuming less power compared to traditional mechanical bell systems. It is user-friendly, as it is easy to install, operate, and maintain even

without advanced technical knowledge. The system can be expanded in the future to control multiple bells or integrated with other devices such as alarms, lights, or notification systems. Since the entire process is automated, the chances of human error, such as missing or delaying the bell, are completely eliminated.

Moreover, the bell sound can be customized according to user preference, and the electronic components used in the system are durable and require less maintenance. With the help of Wi-Fi or Bluetooth, the system can also be monitored or controlled remotely. This project also has educational value, as it helps students and staff understand practical applications of microcontrollers and electronics. Finally, unlike mechanical bells, this system provides a consistent and controlled sound, ensuring quiet and smooth operation.

IX. APPLICATION

- **Schools and Colleges** – Automatically ring bells according to class schedules.
- **Offices** – Schedule notifications or breaks using the bell system.
- **Factories and Workshops** – Signal shift changes or lunch breaks efficiently.
- **Public Institutions** – Libraries, hospitals, or community centers for scheduled alerts.
- **Event Management** – Ring bells for timed events, seminars, or exams.
- **Smart Homes** – Can be integrated for alerts, reminders, or announcements.
- **Research and Learning** – Useful for learning microcontrollers, electronics, and automation projects.

X. RESULT

The Arduino-based college bell system project was successfully completed and **implemented in the college**. The system accurately and reliably **automates the ringing of bells according to the pre-set schedule**, eliminating the need for teachers or staff to ring the bells manually. The RTC (Real-Time Clock) module ensures that all bells ring at the exact scheduled times, maintaining strict adherence to the college timetable.

The automated system **reduces human errors** and ensures consistent operation throughout the day. Users can easily modify the schedule or customize the bell sound/tone, making the system **flexible and user-friendly**. The electronic components used are **durable, energy-efficient, and safe**, ensuring long-term operation with minimal maintenance.

In addition to its practical utility, the project has **educational value**, as it demonstrates the application of microcontrollers, relay modules, and programming for automation. Overall, the project has been successfully implemented in the college as a **fully automated, efficient, and**

accurate bell ringing system, with potential for future expansion to multiple bells or smart alert systems.

XI. CONCLUSIONS

The proposed Arduino-based automatic bell system has been successfully developed and implemented. It offers a reliable and cost-effective automated solution for automating bell operations in educational institutions.

The system eliminates manual effort, improves time accuracy, and enhances operational efficiency. It also demonstrates the practical application of embedded systems and automation. The project can be further enhanced by integrating advanced technologies such as IoT and mobile applications.

XII. UNIQUE CONTRIBUTIONS OF THE PROPOSED SYSTEM COMPARED TO EXISTING PAPERS

Criteria	My Paper	Google Papers (IJRR, IJERT, IJAITE)
Detailed Hardware Description	Arduino UNO, RTC, Relay, Transistor BC548, Diode 1N4007, Resistors, Adapter, Buzzer, Push Buttons – each component explained in detail	Generally only Arduino UNO + RTC + Relay + Buzzer mentioned; no detailed explanation of components
Circuit Diagram and Pin Configuration	Block Diagram, Circuit Diagram, Pin Table (A4, A5 – RTC, D8 – Relay, D9 – Buzzer, D7 – Transistor)	Mostly only block diagrams; pin configuration tables rarely provided
Software Section	Arduino IDE, programming steps, RTC communication, Relay activation, Manual override	Mentions RTC + Arduino programming; but no details about IDE or debugging tools
Advantages and Applications	Schools, Colleges, Offices, Factories, Smart Homes, Event Management – extensive list	Mostly limited to Schools/Colleges
Future Scope	IoT integration, Smart Devices, Multiple Bells, Attendance System Sync, Solar Power, Cloud Logging	Usually only IoT integration or multiple bells mentioned
Educational Value	Useful for students to learn Embedded Systems, Automation, Microcontrollers	Educational value mentioned less frequently
Depth	Step-by-step working principle (Power supply, Initialization, Time monitoring, Relay activation, Manual control, Protection)	Brief description; no step-by-step explanation

XIII. FUTURE SCOPE

The Arduino-based college bell system has immense potential for further development and enhancement. Some of the key future scopes include:

1. **Integration with Smart Devices** – The system can be connected with smartphones or tablets, allowing remote control, notifications, or schedule updates via mobile apps.
2. **IoT-based Automation** – By connecting the system to the Internet of Things (IoT), the bell schedule can be updated remotely, monitored online, and integrated with other smart campus systems.
3. **Multiple Bell Systems** – The project can be extended to control multiple bells across different classrooms, floors, or buildings within the campus.
4. **Customizable Bell Tones and Music** – Different bell tones or melodies can be programmed for specific events, breaks, or special occasions, making the system more engaging.
5. **Integration with Attendance Systems** – The bell system can be synchronized with RFID-based or biometric attendance systems to trigger signals for entry, exit, or class start times.
6. **Energy Efficiency Enhancements** – Use of solar power or low-power electronic components can make the system more energy-efficient and eco-friendly.
7. **Event Scheduling and Alerts** – The system can be adapted for seminar notifications, examination alerts, or emergency announcements in real-time.
8. **Educational Projects and Research** – This project can serve as a base for students to learn about microcontrollers, automation, IoT, and embedded systems through modifications or improvements.
9. **Integration with Visual and Audio Alerts** – Adding LED displays, screens, or speakers can provide visual cues along with bell ringing, helping students with hearing difficulties or providing more interactive notifications.
10. **Cloud Data Logging** – Bell usage data can be stored in the cloud for analysis, helping administrators optimize schedules or monitor system performance.

REFERENCES

1. Dinda RA, Sadrina S, Mursyidin M. The High Accurate Automatic School Bell Controller Based On Arduino Uno DS1307 I2C Real-time Clock. Jurnal Teknik Mesin Mechanical Xplore. 2023 Jul 24;4(1):17-26.
2. Reddy AV, Sharmila V, Chandrasekhar S. Timer Based Automatic College Bell System.

3. Ayson R. Microcontroller-based school bell system for the College of Engineering, DMMMSU-MLUC. DMMMSU Research and Extension Journal. 2021 Dec 1;5:47-66.
4. M. Banzi, Getting Started with Arduino, Maker Media, 2014.
5. S. Monk, Programming Arduino, McGraw-Hill, 2016.
6. Arduino Official Website – https://www.arduino.cc
7. Design and Implementation of Automatic School Bell System Using Microcontroller, International Journal
8. https://www.homemade-circuits.com/automatic-school-college-bell-system-using-arduino/
9. https://www.ijrrjournal.com/IJRR_Vol.11_Issue.11_Nov2024/