

# Project Report



**Course Code:** *EEE 212*

**Course Title:** *Electrical Technology Sessional*

**Submitted To –**

**Md. Naimur Rahman**

Professor

Chairman

Department of Electrical and Electronics Engineering  
Faculty of Computer Science and Engineering.

**Submitted by -**

**Sanjana Tasnim Prome**

ID No. 2002014

Registration No. 09531

Level - 02, Semester - 01

Session - 2020 -2021.

Faculty of Computer Science and Engineering.

**Patuakhali Science and Technology University**  
**Dumki, Patuakhali-8602**

# **Arduino Alarm Clock Project Report**

## **Table of Contents**

1. Introduction
2. Objectives
3. Components Used
4. Circuit Design and Schematic
5. Working Principle
6. Arduino Code Explanation
7. Testing and Results
8. Applications and Future Improvements
9. Conclusion

## **Introduction**

The goal of this project is to build a simple and functional digital alarm clock using an Arduino UNO microcontroller. The clock uses a DS1307 Real-Time Clock (RTC) module to keep track of time accurately and displays the time on a 16x2 LCD display. Users can set the alarm using three push buttons, and a buzzer alerts the user when the alarm is triggered. This project combines hardware and software in an intuitive way to produce a practical time management tool.

## **Objectives**

The main objectives of this project are:

- To create a reliable alarm clock using an Arduino UNO and a DS1307 RTC module.
- To display real-time clock information on a 16x2 LCD display.
- To allow the user to set and adjust the alarm time using push buttons.
- To activate a buzzer when the set alarm time is reached.

## Components Used

1. **Arduino UNO:** The Arduino UNO board is the central microcontroller that handles all the functionalities, including processing input from buttons and controlling the display and buzzer.
2. **DS1307 RTC Module:** The DS1307 Real-Time Clock module provides real-time timekeeping functionality. It has a backup battery to keep the time running even when the Arduino is powered off.
3. **16x2 LCD Display:** The 16x2 LCD display is used to show the current time and the set alarm time. The display can show 2 lines with up to 16 characters on each line.
4. **Push Buttons:** Three push buttons are used to navigate through the time setting options and adjust the hours and minutes of the alarm. One button is for mode selection, and the other two are for incrementing and decrementing values.
5. **Buzzer:** The buzzer sounds an alarm when the set time is reached. It acts as an audible alert for the user.
6. **10K $\Omega$  Potentiometer:** The potentiometer is used to control the contrast of the 16x2 LCD display.

## Circuit Design and Schematic

The circuit is relatively simple and consists of connecting the Arduino UNO to the DS1307 RTC module, the 16x2 LCD, push buttons, and the buzzer.

## Connections

### Arduino UNO to DS1307 RTC Module:

- SDA (Data) pin to Arduino A4
- SCL (Clock) pin to Arduino A5
- VCC to 5V
- GND to GND

**Arduino UNO to LCD (16x2):**

- RS to Arduino Pin 2
- E to Arduino Pin 3
- D4 to Arduino Pin 4
- D5 to Arduino Pin 5
- D6 to Arduino Pin 6
- D7 to Arduino Pin 7
- VSS to GND
- VDD to 5V
- RW to GND
- The potentiometer is connected between the LCD contrast pin and GND.

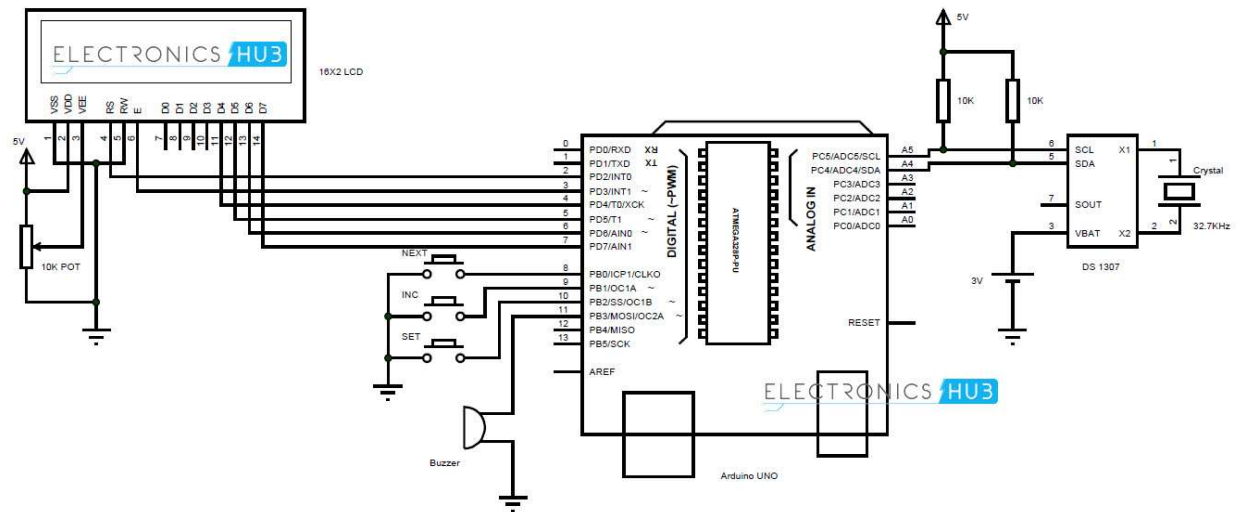
**Push Buttons:**

- Button 1 (Set) to Arduino Pin 10
- Button 2 (Increment) to Arduino Pin 9
- Button 3 (Next) to Arduino Pin 8

**Buzzer:**

- Positive lead of the buzzer to Arduino Pin 11
- Negative lead to GND

## Schematic Diagram:



**Fig 1: Arduino Alarm Clock Circuit Schematic**

## Working Principle

The project functions by continuously reading the time from the DS1307 RTC module and displaying it on the 16x2 LCD display. Users can set the alarm time by pressing the buttons:

1. Button 1 set up the selected time unit (hour or minute).
2. Button 2 increments the selected time unit (hour or minute).
3. Button 3 switches between setting the alarm hour and minutes.

Once the alarm time is set, the clock continuously compares the current time from the RTC module with the alarm time. When both match, the buzzer is activated to alert the user. The buzzer remains on until the alarm is manually reset or stopped.

## Arduino Code Explanation

The Arduino code for this project controls the real-time clock module, LCD, push buttons, and the buzzer. Below are some key segments of the code:

### RTC Setup:

The 'RTCLib' library is used to interface with the DS1307 module.

```
```cpp
#include <Wire.h>

#include <RTCLib.h>

RTC_DS1307 rtc;
```
```

### LCD Initialization:

The 'LiquidCrystal' library is used to control the 16x2 LCD display.

```
```cpp
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```
```

### Button Functionality:

The code reads the state of each push button and adjusts the time or alarm settings accordingly.

```
```cpp

int buttonState1 = digitalRead(buttonPin1);

int buttonState2 = digitalRead(buttonPin2);

int buttonState3 = digitalRead(buttonPin3);
```
```

### Alarm Trigger:

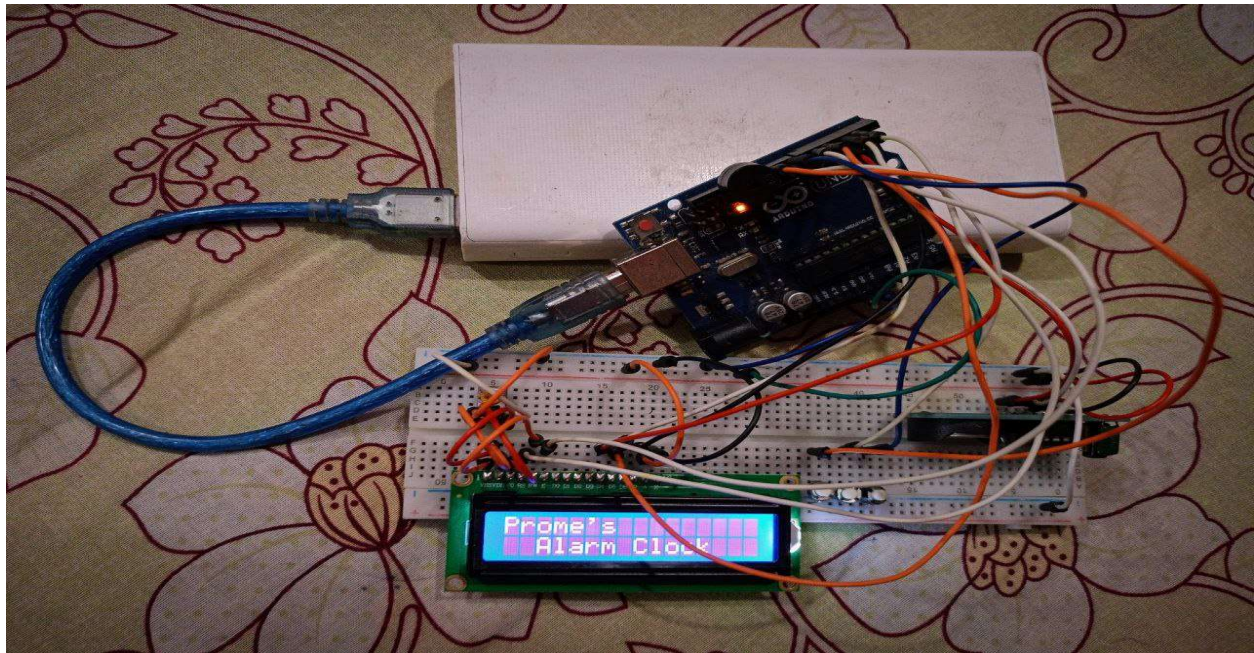
The comparison of the current time to the alarm time to activate the buzzer:

```
```cpp
if (currentTime.hour() == alarmHour && currentTime.minute() == alarmMinute) {
    digitalWrite(buzzerPin, HIGH); // Buzzer ON
}
```
```

### Testing and Results

After assembling the circuit and uploading the code to the Arduino UNO, the following tests were performed to verify the functionality of the system:

- Time Display: The RTC module accurately displays the current time on the 16x2 LCD display.
- Alarm Setting: The push buttons successfully allow the user to set and adjust the alarm time.
- Alarm Trigger: When the current time reaches the set alarm time, the buzzer sounds, signaling the alarm.
- Buzzer Control: The buzzer turns off manually after the user resets the alarm or changes the time.



**Fig 2: Alarm Clock is working**

Results indicate that the Arduino Alarm Clock functions as intended, providing a reliable alarm system.

#### **Cost Estimation**

| Name of Components     | Tk (BDT) |
|------------------------|----------|
| Arduino UNO            | 550/-    |
| RTC DS1307 Module      | 330/-    |
| 16*2 LCD Display       | 360/-    |
| Buzzer                 | 25/-     |
| Push Buttons           | 15/-     |
| Potentiometer 10 K Ohm | 10/-     |
| Jumper Wires           | 30/-     |
| Breadboard             | 80/-     |
| Others                 | 100/-    |
| Total                  | 1500/-   |

**Components from RoboticsBD**



## **Applications and Future Improvements**

### Applications:

- This project can be used as a functional alarm clock in homes or offices.
- It can be integrated into larger IoT systems where time-keeping and alarms are required.

### Future Improvements:

- Snooze Functionality: A snooze button could be added to allow the user to silence the alarm for a few minutes.
- Multiple Alarms: Additional memory could be used to store multiple alarm settings.
- Low-Power Mode: Implementing a low-power mode to extend battery life when the system is idle.

## **Conclusion**

The Arduino-based alarm clock project demonstrates how simple components can be used to build a functional and customizable time management system. The use of the DS1307 RTC ensures accurate timekeeping, while the LCD display provides a user-friendly interface. With further improvements, this project could serve as a practical tool in daily life.