

Project Proposal



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-Based Digital Alarm Clock

Introduction

In today's fast-paced world, accurate time management is essential. Alarm clocks are crucial for helping people stay punctual and organized. Traditional alarm clocks, though widely used, often lack customization options and advanced features like real-time clock accuracy or user-friendly interfaces.

This project aims to design and build a digital alarm clock using an Arduino UNO, which integrates a DS1307 Real-Time Clock (RTC) module, a 16x2 LCD display, and push buttons. By using an Arduino-based system, this project offers flexibility, expandability, and an easy-to-program solution. The system will allow users to set alarms with a precise clock, show the time on an LCD screen, and trigger a buzzer at the alarm time.

This proposal outlines the project scope, objectives, methodology, and the expected outcomes of creating a cost-effective and reliable digital alarm clock.

Objectives

The key objectives of the project are as follows:

1. Accurate Time Display: Utilize the DS1307 RTC module to provide accurate timekeeping, even during power interruptions.
2. User-Friendly Interface: Display the time and alarm settings on a 16x2 LCD display. Three push buttons will allow the user to set the alarm time easily.
3. Alarm Functionality: When the alarm time is reached, the system will activate a buzzer as an audible alert for the user.
4. Customizable and Expandable: The Arduino platform allows for future expansion, such as adding additional alarm times, snooze functions, or integrating other features like temperature sensors.

Methodology

The development of the Arduino Alarm Clock will follow a systematic approach. The process is broken down into the following phases:

Component Selection and Procurement

The first step involves gathering all required components for the project. The main components include:

- Arduino UNO
- DS1307 Real-Time Clock (RTC) Module
- 16x2 LCD Display
- 3 Push Buttons
- 1 Buzzer
- 10 K Ω Potentiometer (POT)

These components will be sourced from reliable suppliers to ensure quality.

Circuit Design and Schematic

Using circuit design software (like Fritzing), the wiring diagram will be created to show how the components are connected to the Arduino. The RTC module will interface with the Arduino using I2C communication, and the LCD display will be connected in 4-bit mode to reduce the number of required pins

The push buttons will be used for selecting, incrementing, and decrementing the alarm time, while the potentiometer will adjust the contrast of the LCD display.

Software Development

Arduino IDE will be used to write and upload code to the Arduino board. The code will:

- Read and display real-time data from the DS1307 RTC module.
- Allow the user to set an alarm time using push buttons.
- Trigger the buzzer when the current time matches the alarm time.
- Update the LCD display with real-time information.

The software will use the `RTClib` library for communicating with the DS1307 module and the `LiquidCrystal` library for controlling the LCD display.

Assembly and Integration

Once the circuit and code are developed, the components will be assembled on a breadboard or prototype board for initial testing. The Arduino, RTC module, LCD display, push buttons, and buzzer will be connected as per the schematic, and initial tests will be conducted to verify functionality.

Testing and Debugging

The system will undergo multiple rounds of testing to ensure that:

- The RTC module accurately tracks time.
- The LCD correctly displays the time and alarm settings.
- The push buttons reliably set the alarm.
- The buzzer activates when the alarm time is reached.

During testing, potential issues like component failure, incorrect wiring, or software bugs will be addressed through debugging.

Expected Outcomes

Upon successful completion of the project, the following outcomes are expected:

Working Alarm Clock: The system will display accurate time, allow users to set an alarm, and trigger a buzzer at the appropriate time.

User-Friendly Interface: The use of a 16x2 LCD display and push buttons will create a simple and intuitive interface for setting and adjusting the alarm.

Expandable System: The project can be further expanded with additional features like multiple alarms, snooze functionality, or integration into a smart home system.

Timeline

The project is expected to take approximately 4-5 weeks, with the following breakdown:

Phase	Duration	Expected Completion
Component Selection and Procurement	1 Week	1 Week
Circuit Design and Schematic	1 Week	2 Week
Software Development	1 Week	3 Week
Assembly and Integration	1 Week	4 Week
Testing and Debugging	1 Week	5 Week

Conclusion

The Arduino-based alarm clock project is a cost-effective, user-friendly solution for creating a reliable and customizable alarm system. Leveraging Arduino's flexibility and the DS1307 RTC's precision, this project provides a solid foundation for further enhancements, such as additional alarms, snooze functionality, and even integration into larger home automation systems.

With a clear plan, achievable objectives, and a well-structured timeline, this project will offer valuable insights into embedded systems and microcontroller-based design.

This proposal covers all essential aspects of your project. Feel free to adjust or elaborate on any sections to suit your specific requirements!