Development of an Arduino-Powered Digital Clock with Timer Capability

Arnav M - EE24BTECH11006

March 25, 2025

Abstract

This document outlines the creation of an Arduino-controlled digital clock that integrates a timer function. The design utilizes a 7447 BCD-to-7-segment decoder in conjunction with common anode (active-low) 7-segment displays. User interaction is facilitated through push buttons, while precise timekeeping is achieved using a timer interrupt. Multiplexing is employed to efficiently manage six display digits representing hours, minutes, and seconds.

1 Introduction

Digital clocks serve as fundamental components in embedded systems. This project implements an Arduino-driven timekeeping device capable of displaying hours, minutes, and seconds, while also featuring a countdown timer mode. The design incorporates a 7447 decoder for efficient digit control and active-low common anode displays. The clock operates using a multiplexed display strategy and allows user adjustments via pushbutton inputs.

2 System Components

2.1 Hardware Elements

The system consists of the following hardware components:

- **Arduino Microcontroller:** Responsible for time management, input processing, and controlling display operations.

- **7447 BCD-to-7-Segment Decoder:** Converts binary-coded decimal (BCD) input from the Arduino into signals to drive the 7-segment display.
- **Six Common Anode 7-Segment Displays:** Active-low displays used for showing the time in HH:MM:SS format.
- **Five Push Buttons:** Used to increment time, switch modes, and reset values.
- **Resistors:** Six 220 Ω resistors protect the displays by limiting current.

2.2 Circuit Connections

2.2.1 Interfacing BCD with the 7447 Decoder

The Arduino outputs BCD signals via digital pins PD2 through PD5, directing them to the 7447 decoder. As the displays are common anode, the decoder outputs operate in active-low mode. The wiring details are provided in Table

Arduino Pin	7447 BCD Input
PD2	A
PD3	B
PD4	C
PD5	D

Table 1: BCD Inputs to 7447 Decoder

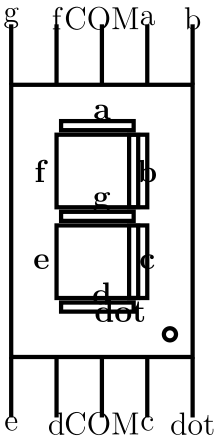

Figure 1: Interfacing Seven Segment Display with Arduino

2.2.2 Multiplexing Display Digits

To efficiently control all six digits with a limited number of Arduino pins, a multiplexing approach is implemented. This technique allows the Arduino to activate each digit sequentially at high speed, creating the illusion of a continuously lit display. Table

Digit Position	Arduino Pin
Seconds (Units)	6
Seconds (Tens)	7
Minutes (Units)	8
Minutes (Tens)	9
Hours (Units)	10
Hours (Tens)	11

Table 2: 7-Segment Display Digit Selection Pins

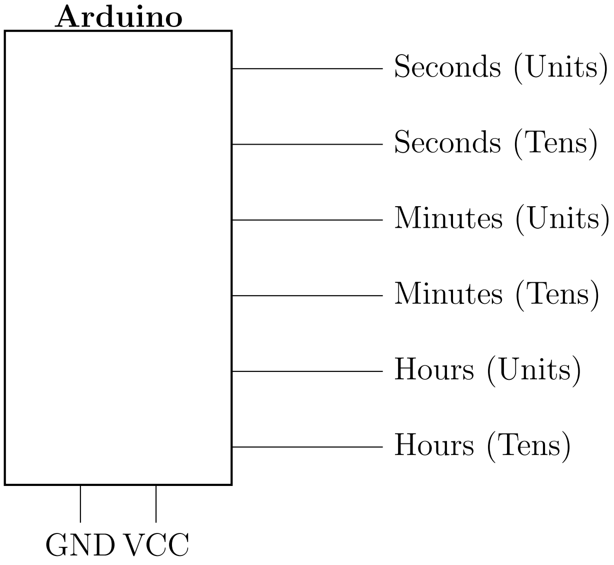

Figure 2: Multiplexing Display Digits with Arduino

3 Software Design and Implementation

3.1 System Operation

The Arduino-based system operates in two modes:

- **Clock Mode:** Displays hours, minutes, and seconds, incrementing the time every second.
- **Timer Mode:** Allows the user to set a countdown timer, decrementing the time until it reaches zero.

The system switches between these modes using a dedicated button. Upon completion of the countdown, the display blinks to indicate the end of the timer.

3.2 Time Management Using Interrupts

Timer interrupts ensure precise timekeeping. A 1ms tick is generated using **Timer0** configured in CTC (Clear Timer on Compare Match) mode. Time increments occur every 1000ms, and the system updates the display accordingly.

3.3 User Interaction via Push Buttons

Five push buttons provide user interaction:

- **Increment Buttons:** Used to adjust hours, minutes, and seconds.
- **Mode Button:** Switches between clock and timer modes.
- **Pause/Resume Button:** Pauses and resumes the clock or timer.
- **Reset Button:** Long press resets the clock or timer values.

3.4 Error Handling and Safety Measures

Boundary conditions, such as exceeding 24 hours or 60 minutes, are automatically handled by resetting to zero. Button debouncing techniques prevent unintended multiple triggers.

4 Conclusion

This project successfully implements a digital clock with an integrated timer, leveraging a 7447 BCD decoder and multiplexed display control. Timer interrupts ensure precise timekeeping, and push buttons facilitate user interaction. The system efficiently switches between clock and timer modes while providing visual feedback through blinking when the timer completes.

The project serves as an excellent demonstration of embedded system design principles. The code is available in the following git hub link: `https://github.com/arnavmahishi/EE1003/tree/main/hardware`