Arduino-Based Digital Clock with Timer Mode

Dwarak A - EE24BTECH11019

March 24, 2025

Abstract

This report describes the design and implementation of an Arduino-based digital clock with an integrated timer mode. The system utilizes a 7447 BCD to 7-segment decoder with common anode (active low) 7-segment displays and button-based controls for time adjustments. Timekeeping is managed via a timer interrupt, while the display is driven using multiplexing to handle six digits representing hours, minutes, and seconds.

1 Introduction

Digital clocks play a crucial role in embedded systems. This project implements an Arduino-based clock that displays time in hours, minutes, and seconds while providing a countdown timer mode. Key features include the use of a 7447 BCD to 7-segment decoder (designed for common anode displays, meaning the segment outputs are active low), multiplexed display control, and several pushbuttons for user interaction.

2 Hardware Setup

2.1 Components Required

Component	Quantity
Arduino Board	1
7447 BCD to 7-Segment Decoder	1
Common Anode 7-Segment Displays	6
Push Buttons	5
Resistors (220 Ω)	6
Connecting Wires	As required
Breadboard	1
Power Supply (5V)	1

Table 1: List of Required Components

2.2 Basic Connections

2.2.1 BCD Outputs to 7447 Decoder

The Arduino outputs BCD values on digital pins PD2–PD5, which are fed into the 7447 decoder. Since the displays are common anode, the decoder outputs are active low. The connections are shown in Table 2.

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

Table 2: BCD Outputs to 7447 Decoder

2.2.2 Multiplexed 7-Segment Displays

To display six digits using limited Arduino pins, multiplexing is employed. The Arduino sequentially activates each digit by driving its enable pin. The digit connections are listed in Table 3.

2.2.3 Pushbutton Connections

The pushbuttons, connected using the Arduino's internal pull-up resistors, pull the corresponding input LOW when pressed. Their connections are given in Table 4.

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

Table 3: Multiplexed 7-Segment Display Digit Enable Connections

Function	Arduino Pin
Hour Increment	A0
Minute Increment	A1
Second Increment	A2
Pause/Play Toggle	A3
Mode Toggle (Clock/Timer)	A5

Table 4: Pushbutton Connections

3 Arduino to AVR Pin Mapping

The following table shows how the Arduino board's pin labels correspond to the AVR microcontroller's port names as referenced in AVR GCC. This mapping is essental for low-level programming and direct port manipulation.

Arduino Pin	**AVR Pin/Port**
D2	PD2
D3	PD3
D4	PD4
D5	PD5
D6	PD6
D7	PD7
D8	PB0
D9	PB1
D10	PB2
D11	PB3
A0	PC0
A1	PC1
A2	PC2
A3	PC3
A5	PC5

Table 5: Mapping of Arduino Pins to AVR GCC Pin Terms

4 IC and Display Pinouts

4.1 7-Segment Display Pinout

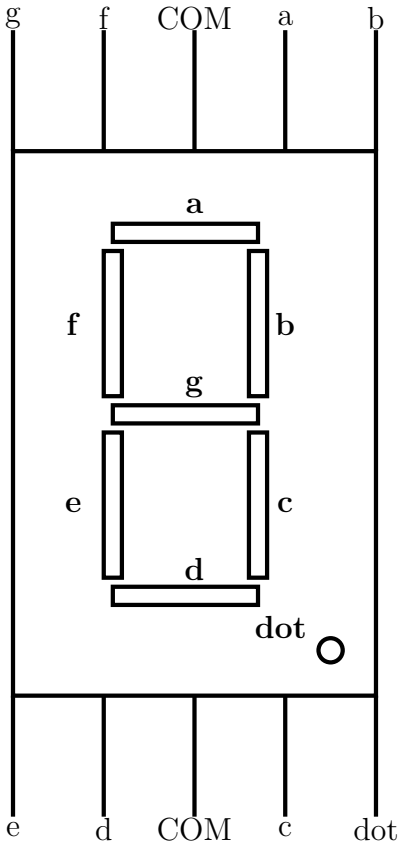

Figure 1: 7-Segment Display Pinout (Common Anode, Active Low)

4.2 7447 Decoder Pinout

Below is the placeholder for your TikZ picture for the 7447 decoder pinout. Insert your TikZ code here.

5 Multiplexing Explanation

Multiplexing is employed to drive multiple 7-segment displays using a limited number of Arduino pins. The process is as follows:

1. The Arduino sets the BCD outputs (PD2–PD5) to the binary value corresponding to the digit to be displayed.

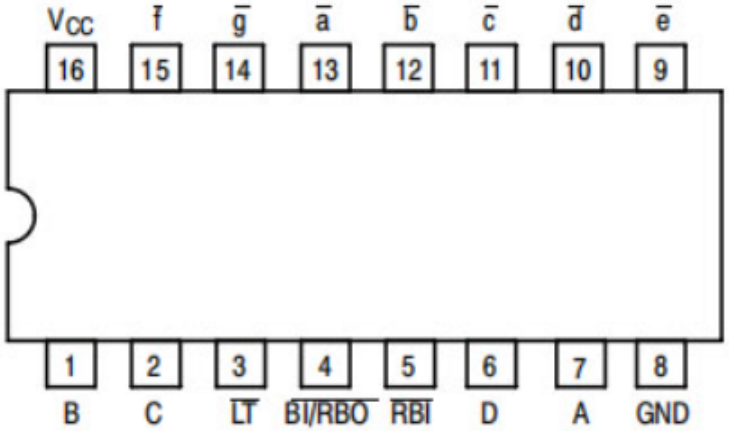

Figure 2: 7447 IC Pinout Diagram

2. It then activates the corresponding digit enable pin (one of pins 6–11) for a brief period (of the order 10^{-3} s).

3. The digit is then deactivated, and the Arduino moves to the next digit.

This rapid cycling (typically at least 50 Hz per digit) creates the illusion of all digits being lit simultaneously, due to persistence of vision.

6 Timekeeping with Timer Interrupt

The Arduino utilizes Timer0 to generate an interrupt every 1 ms. This interrupt increments a global millisecond counter. When the counter reaches 1000, indicating one second has elapsed, the following steps occur:

- The seconds value is incremented.
- When seconds reach 60, they reset to 0 and the minutes counter is incremented.
- Similarly, when minutes reach 60, the hours counter is incremented.
- When hours reach 24, the clock resets to 00:00:00.

7 Button Functions

7.1 Increment Buttons (A0–A2)

- **Short press**: Increments the corresponding time unit (hour, minute, or second).
- **Long press (after 2 seconds)**: Initiates continuous incrementing at a rate of one increment every 200 ms.

7.2 Pause/Play Button (A3)

- **Short press**: Toggles between pause and play states.
- **Long press (after 5 seconds)**: Resets the active mode's time to 00:00:00 and pauses the clock.

7.3 Clock/Timer Mode Toggle (A5)

Pressing the mode toggle button switches between clock mode (count-up) and timer mode (count-down) without losing the stored time for either mode.

8 Source Code

For easy access to the complete source code of this project, visit the following repository:

`https://github.com/Dwarak-A/EE1003/blob/main/hardware/clock`

9 Conclusion

This project demonstrates the implementation of an Arduino-based digital clock featuring a timer mode. By utilizing a 7447 BCD to 7-segment decoder with common anode displays (active low) and employing multiplexing, the design efficiently drives a six-digit display. The use of timer interrupts ensures accurate timekeeping, and the system allows user interaction through multiple pushbuttons.