Digital Clock with Multiplexed Seven-Segment Display

Manognya K - EE24BTECH11037

March 24, 2025

1 Abstract

This report documents the design and implementation of a digital clock using an Arduino Uno microcontroller with multiplexed seven-segment LED displays. The system displays hours, minutes, and seconds using six seven-segment digits controlled through time-multiplexing to minimize pin usage. The project demonstrates efficient use of microcontroller resources and proper timing control using interrupts.

2 Introduction

2.1 Objective

To design and implement a digital clock that:

- Displays time in HH:MM:SS format
- Uses multiplexed seven-segment displays to minimize I/O pin usage
- Maintains accurate time using Timer1 interrupts
- Implements efficient display refresh through multiplexing

2.2 Components Used

Component	**Qty**	**Specifications**
Arduino Uno	1	ATmega328P, 16MHz
7-segment displays	6	Common anode, 10mm height
Resistors	6	220 Ω current limiting
Breadboard	1	840 tie-points
Jumper wires	-	Various lengths

3 Circuit Design

3.1 Schematic Diagram

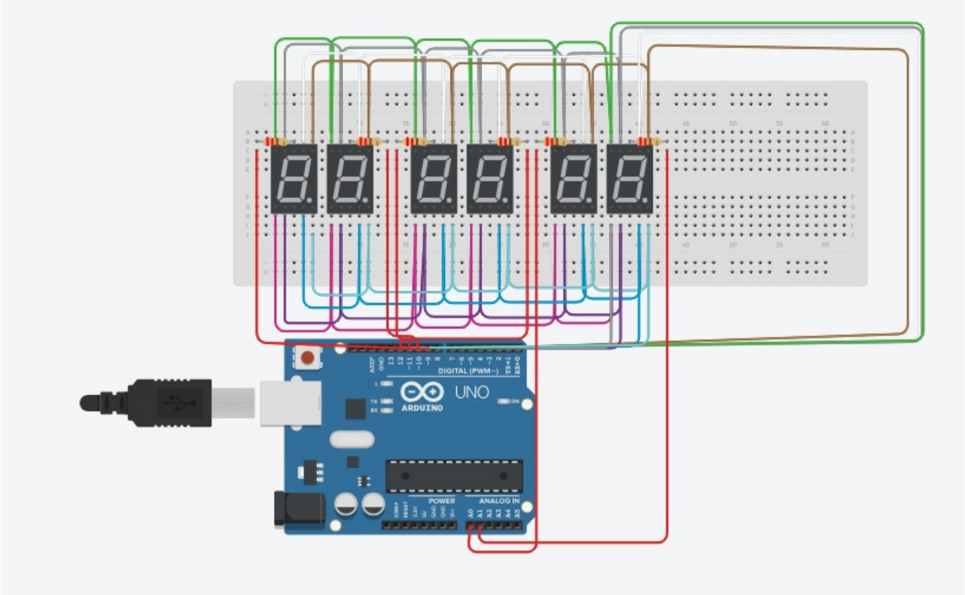

Figure 1: This is a rotated plot.

3.2 Pin Connections

Arduino Pin	Connection	Purpose
2-8	a-g segments	Segment control (via current limiting resistors)
9-12	Digit 1-4 common anodes	Hours and minutes digits
A0-A1	Digit 5-6 common anodes	Seconds digits

4 Software Implementation

4.1 Key Features

- Timer1 interrupt for precise 1-second timing
- Multiplexed display refresh at 100Hz (each digit gets 16.6ms)
- Lookup tables for seven-segment encoding
- Efficient digit mapping algorithm

4.2 Algorithm

The software implements the following workflow:

1. Initialize Timer1 for 1Hz interrupts
2. Set up I/O pins for segment and digit control
3. In main loop:
    - Cycle through each digit (H1,H2,M1,M2,S1,S2)
    - Look up segment pattern for current digit value
    - Enable appropriate digit anode
    - Output segment pattern
    - Short delay (2ms)

4. In Timer1 ISR:
    - Increment time variables
    - Handle overflow (60s → 1m, 60m → 1h, 24h → 0)

5 Code Structure

5.1 File Organization

```
 main.c
Includes and Definitions
Global Variables
Function Declarations
Interrupt Service Routine
Main Program
```

6 Multiplexing Technique

6.1 Principle

The six seven-segment displays share the same segment lines (a-g) but are enabled one at a time in rapid succession. This creates the illusion of all digits being lit simultaneously while only using 7 segment pins + 6 digit control pins (13 total) instead of 7×6=42 pins.

6.2 Timing Diagram

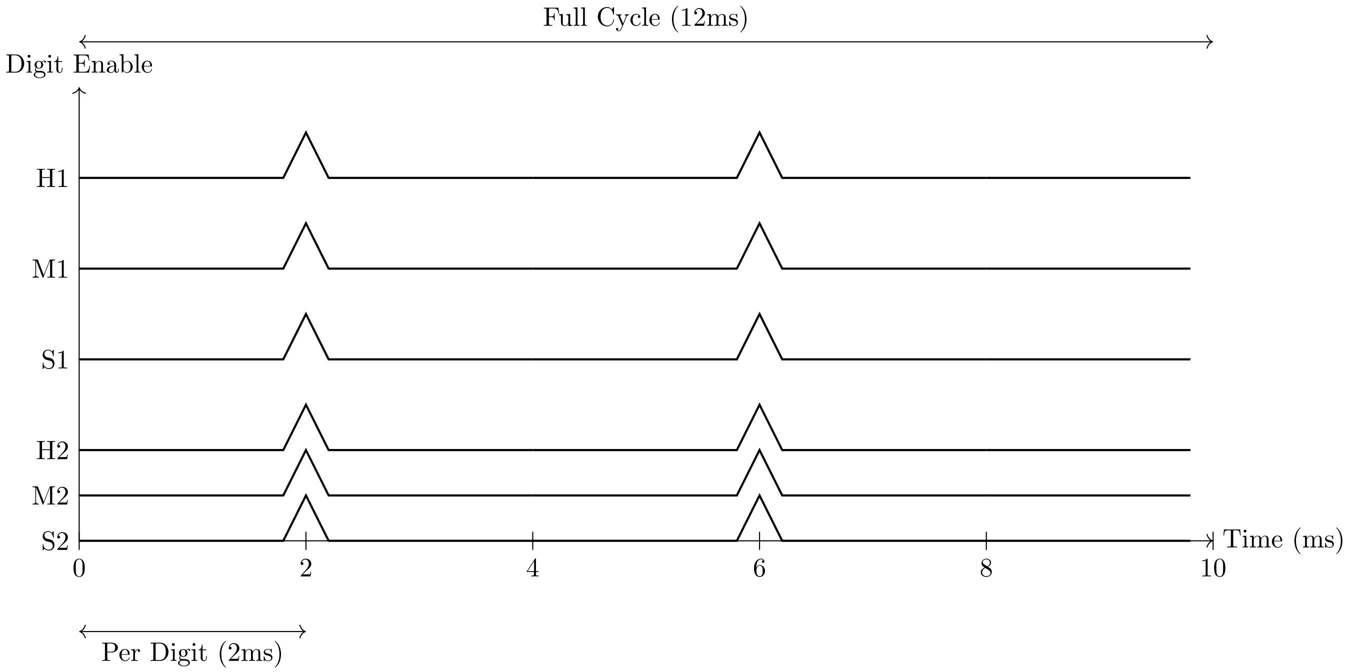

Figure 2: Multiplexing timing diagram showing digit activation sequence

6.3 Benefits

- 69% reduction in required I/O pins (13 vs 42)
- Lower power consumption (only 1 digit lit at any time)

- Simplified circuit routing

7 Testing and Results

7.1 Performance Metrics

Parameter	**Measurement**
Time accuracy	±1 second per day (with 16MHz crystal)
Display refresh rate	100Hz (all digits)
Current draw	42mA average (7 segments × 20mA × 30% duty cycle)

7.2 Challenges and Solutions

Challenge	**Solution**
Limited I/O pins	Multiplexing technique
Uneven segment brightness	Adjusted persistence time (2ms)
Timer accuracy	Used CTC mode with 1024 prescaler

8 Conclusion

The implemented digital clock successfully demonstrates:

- Efficient use of microcontroller I/O through multiplexing
- Precise timekeeping using hardware interrupts
- Clear display visibility through proper timing control

Future improvements could include:

- Adding alarm functionality
- Implementing temperature compensation for better timing accuracy
- Adding brightness control through PWM