Digital Clock Implementation using Arduino with Multiplexing and Editing Features

Dhawal
Department of Electrical Engineering
Indian Institute of Technology Hyderabad
Email: ee24btech11015@iith.ac.in

I. Introduction

The digital clock system described here implements a feature-rich clock with editing capabilities using an Arduino microcontroller. The system utilizes a multiplexing technique to display time on six seven-segment displays using minimal I/O pins. This implementation includes pause/play functionality and digit-by-digit editing with increment and decrement buttons.

II. Components

Component	Value	Quantity
Arduino Uno		1
USB Cable	Type B	1
Seven Segment Display	Common Cathode	6
Push Buttons		4
IC 7447		1
Jumper Wires	M-M	16
Breadboard		1
Resistors	220Ω	7
Resistors	$10k\Omega$ (pull-down)	4

Table 1.0: Components List

III. Circuit Connections

A. Connections to Arduino

Make the button connections and IC 7447 connections to the Arduino as per the table below.

Item	Arduino Pin	Function
Button 1	A0 (PC0)	Edit Mode Toggle
Button 2	A1 (PC1)	Next Digit Selection
Button 3	A2 (PC2)	Increment Digit
Button 4	A3 (PC3)	Decrement Digit
IC 7447 Pin 7	D2	BCD Bit 0 (LSB)
IC 7447 Pin 1	D3	BCD Bit 1
IC 7447 Pin 2	D4	BCD Bit 2
IC 7447 Pin 6	D5	BCD Bit 3 (MSB)
Display 1	D6	Hours Tens Digit
Display 2	D7	Hours Units Digit
Display 3	D8	Minutes Tens Digit
Display 4	D9	Minutes Units Digit
Display 5	D10	Seconds Tens Digit
Display 6	D11	Seconds Units Digit

Table 2.0: Arduino Pin Connections

B. Connections from Seven Segment to BCD

Make the seven-segment connections identical for all seven segments. In total, there should only be 7 wires of output coming from the seven-segment display array.

IC 7447	Seven Segment (All)	Name
Pin 13	a	Controls segment a
Pin 12	b	Controls segment b
Pin 11	c	Controls segment c
Pin 10	d	Controls segment d
Pin 9	e	Controls segment e
Pin 15	f	Controls segment f
Pin 14	g	Controls segment g
Pin 8	Ground	Ground Supply
Pin 16	5V	Power Supply

Table 3.0: BCD to 7-Segment Connections

IV. Multiplexing Technique

Multiplexing is achieved by connecting all segment inputs of the seven-segment displays to a single BCD decoder. Digital pins are connected to the common cathode of each display, allowing selective activation of each display to show the BCD output. The displays are alternated with a very small time gap (2ms), creating the illusion of simultaneous operation.

V. Digit Editing Logic

The clock features a comprehensive editing system that allows precise time adjustment:

1) Button 1 toggles between run mode and edit mode
2) In edit mode, Button 2 cycles through the six digits (hours tens, hours units, minutes tens, etc.)
3) Button 3 increments the selected digit with proper rollover constraints
4) Button 4 decrements the selected digit with proper rollunder constraints

The editing logic applies different constraints based on digit position. The following tables show the Boolean logic for incrementing and decrementing each digit type:

TABLE I
INCREMENT LOGIC FOR UNITS DIGITS (0-9)

D	C	B	A	D_1	C_1	B_1	A_1
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

A. Increment Logic for Units Digits (0-9)

Simplified Boolean expressions for incrementing units digits:

$$A_1 = A' \tag{1}$$

$$B_1 = A + B \tag{2}$$

$$C_1 = AB + C \tag{3}$$

$$D_1 = ABC + D \tag{4}$$

B. Increment Logic for Tens of Minutes/Seconds (0-5)

TABLE II
INCREMENT LOGIC FOR TENS OF MINUTES/SECONDS (0-5)

D	C	B	A	D_1	C_1	B_1	A_1
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	0	0	0

Simplified Boolean expressions for incrementing tens of minutes/seconds:

$$A_1 = A'B'C' + AB'C \tag{5}$$

$$B_1 = A'BC' + AB'C' \tag{6}$$

$$C_1 = A'BC + AB'C \tag{7}$$

$$D_1 = 0 \tag{8}$$

C. Increment Logic for Tens of Hours (0-2)

TABLE III
INCREMENT LOGIC FOR TENS OF HOURS (0-2)

D	C	B	A	D_1	C_1	B_1	A_1
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0

Simplified Boolean expressions for incrementing tens of hours:

$$A_1 = A'B' \tag{9}$$

$$B_1 = AB' \tag{10}$$

$$C_1 = 0 \tag{11}$$

$$D_1 = 0 \tag{12}$$

TABLE IV
DECREMENT LOGIC FOR UNITS DIGITS (0-9)

D	C	B	A	D_1	C_1	B_1	A_1
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	1
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	1
0	1	1	1	0	1	1	0
1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	0

D. Decrement Logic for Units Digits (0-9)

Simplified Boolean expressions for decrementing units digits:

$$A_1 = A \tag{13}$$

$$B_1 = A' + B \tag{14}$$

$$C_1 = A'B' + C \tag{15}$$

$$D_1 = A'B'C' + D \tag{16}$$

E. Decrement Logic for Tens of Minutes/Seconds (0-5)

TABLE V
DECREMENT LOGIC FOR TENS OF MINUTES/SECONDS (0-5)

D	C	B	A	D_1	C_1	B_1	A_1
0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	0	1	3	0	0	1	0
0	1	0	0	0	0	1	1
0	1	0	1	0	1	0	0

Simplified Boolean expressions for decrementing tens of minutes/seconds:

$$A_1 = A'B'C + AB'C' \tag{17}$$

$$B_1 = A'BC' + AB'C' \tag{18}$$

$$C_1 = A'B'C' + ABC' \tag{19}$$

$$D_1 = 0 \tag{20}$$

F. Decrement Logic for Tens of Hours (0-2)

TABLE VI
DECREMENT LOGIC FOR TENS OF HOURS (0-2)

D	C	B	A	D_1	C_1	B_1	A_1
0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1

Simplified Boolean expressions for decrementing tens of hours:

$$A_1 = BA' \tag{21}$$

$$B_1 = B'A' \tag{22}$$

$$C_1 = 0 \tag{23}$$

$$D_1 = 0 \tag{24}$$

VI. Control Implementation

1) Pressing Button 1 toggles between run mode and edit mode. In edit mode, the clock pauses.
2) In edit mode, pressing Button 2 selects the next digit for editing (cycles through all six digits).
3) In edit mode, pressing Button 3 increments the currently selected digit using the increment logic tables.
4) In edit mode, pressing Button 4 decrements the currently selected digit using the decrement logic tables.
5) The selected digit blinks at 5Hz (200ms on, 200ms off) for visual feedback.

VII. Software Implementation

The Arduino code implements:

- Timer interrupt for clock ticking (10Hz interrupt rate)
- Button debouncing with software delays
- Multiplexed display refresh
- Editing mode with digit selection and value modification using the Boolean logic from the tables
- Proper constraints on time values (hours 0-23, minutes 0-59, seconds 0-59)

VIII. Execution

A. Upload Code to Arduino

1) Connect Arduino to computer via USB
2) Open Arduino IDE
3) Copy the provided code into a new sketch
4) Select the correct board and port
5) Upload the code

B. Hardware Build

- Connect the seven-segment displays to the breadboard
- Connect all segment outputs together (through resistors)
- Make connections to the IC7447 according to Table 3.0
- Connect the IC7447 and the buttons to the Arduino according to Table 2.0
- Add appropriate current-limiting resistors for LEDs and pull-down resistors for buttons

Acknowledgment

The complete source code and documentation can be found at: https://github.com/Dhawal24112006/projects.git

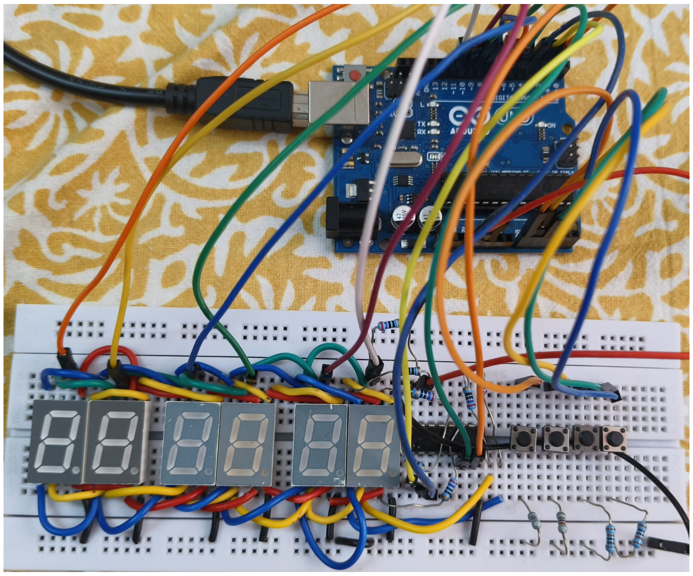

Fig. 1. Final Arduino-based Clock Implementation

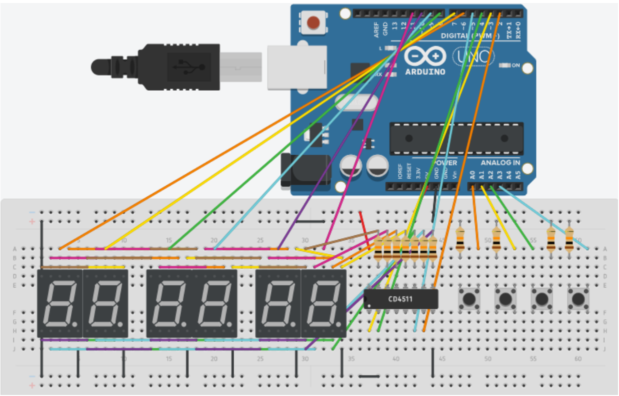

Fig. 2. Tinkercad Simulation of the Digital Clock