

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

Abstract—In this paper the design and implementation of a feature-rich digital clock is demonstrated. The system uses multiplexing to drive six seven-segment displays efficiently, minimizing I/O utilization. Key functionalities include timekeeping, digit-by-digit editing, and pause/play control. Boolean-based increment and decrement logic ensures more accurate cascading of seconds, minutes, and hours within standard constraints. The hardware setup, complemented by software debouncing and display refreshing, demonstrates a reliable, compact, and user-interactive digital clock suitable for both educational and practical applications.

I. INTRODUCTION

Digital timekeeping has long been a critical component of electronic system design, with classical digital design principles thoroughly discussed in foundational works such as [1], [2], and [3]. The advent of microcontroller platforms, particularly Arduino, has enabled the development of compact, programmable clocks with enhanced user interactivity [4]. Techniques such as BCD-to-seven-segment interfacing and display multiplexing allow efficient utilization of limited I/O resources while maintaining accurate visual representation [5]. Inspired by these principles, this work presents an Arduino-based digital clock featuring six-digit multiplexed displays, pause/play functionality, and digit-by-digit editing with Boolean logic-driven increment and decrement operations.

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	220 Ω	7
Resistors	10k Ω (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	D10	Edit Mode Toggle
Button 2	D11	Next Digit Selection
Button 3	D12	Increment Digit
Button 4	D13	Decrement Digit
IC 7447 Pin 7	D0	BCD Bit 0 (A)
IC 7447 Pin 1	D1	BCD Bit 1 (B)
IC 7447 Pin 2	D2	BCD Bit 2 (C)
IC 7447 Pin 6	D3	BCD Bit 3 (D)
Display 1	D4	Hours Tens Digit
Display 2	D5	Hours Units Digit
Display 3	D6	Minutes Tens Digit
Display 4	D7	Minutes Units Digit
Display 5	D8	Seconds Tens Digit
Display 6	D9	Seconds Units Digit

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

All BCD inputs (A-D) are shared among six seven-segment displays. Displays are enabled one at a time using EN[0..5] = D4-D9. Each digit is displayed for 1ms, creating a fast alternating effect that appears continuous. This saves I/O pins and allows full six-digit display.

V. DIGIT EDITING LOGIC

The clock allows pausing and digit-by-digit editing:

- 1) Press PAUSE (D10) to toggle run/edit mode. In edit mode, the clock stops.

- 2) Press NEXT (D11) to select the digit to edit (cycles 0-5: sec1, sec10, min1, min10, hr1, hr10).
- 3) Press INC (D12) to increment the selected digit with rollovers.
- 4) Press DEC (D13) to decrement the selected digit with rollunders.
- 5) Selected digit blinks every 500ms to indicate focus.

VI. CONSTRAINTS EXPLANATION

- **Seconds and Minutes Ones:** 0–9, standard BCD.
- **Seconds and Minutes Tens:** 0–5, to match 0–59 range.
- **Hours Ones:** 0–9 if hours tens = 0 or 1, but 0–3 if hours tens = 2, ensuring 24-hour format.
- **Hours Tens:** 0–2.

VII. INCREMENT LOGIC AND TRUTH TABLES

A. Seconds Ones (0-9)

Z	Y	X	W	D	C	B	A
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

ZY \ XW	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	-	-	-	-
10	1	0	-	-

$$A = W'_1$$

ZY \ XW	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	-	-	-	-
10	0	0	-	-

$$B = (W_1 X'_1 Z'_1) + (W'_1 X_1)$$

ZY \ XW	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	-	-	-	-
10	0	0	-	-

$$C = (X'_1 Y_1) + (W'_1 Y_1) + (W_1 X_1 Y'_1)$$

ZY \ XW	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	-	-	-	-
10	1	0	-	-

$$D = (W'_1 Z_1) + (W_1 X_1 Y_1)$$

B. Seconds Tens (0-5)

Z	Y	X	W	D	C	B	A
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

ZY \ XW	00	01	11	10
00	1	0	0	1
01	1	0	-	-
11	-	-	-	-
10	-	-	-	-

$$A = W'_2$$

ZY \ XW	00	01	11	10
00	0	1	0	1
01	0	0	-	-
11	-	-	-	-
10	-	-	-	-

$$B = (W_2 X'_2 Y'_2) + (W'_2 X_2)$$

ZY \ XW	00	01	11	10
00	0	0	1	0
01	1	0	-	-
11	-	-	-	-
10	-	-	-	-

$$C = (W_2 X_2) + (W'_2 X'_2 Y_2)$$

$$D = 0$$

C. Minutes Ones (0-9)

Same as Seconds Ones with W3/X3/Y3/Z3.

D. Minutes Tens (0-5)

Same as Seconds Tens with W4/X4/Y4/Z4.

E. Hours Ones

I. Tens = 0/1 → 0-9

Same as Seconds Ones with W5/X5/Y5/Z5.

II. Tens = 2 → 0-3

X	W	D	C	B	A
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
1	1	0	0	0	0

X \ W	0	1
0	1	0
1	1	0

$$A = W'_5$$

X \ W	0	1
0	0	1
1	1	0

$$B = (W_5 X'_5) + (W'_5 X_5)$$

$$C = 0$$

$$D = 0$$

F. Hours Tens (0-2)

X	W	D	C	B	A
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	0	0	0

X \ W	0	1
0	1	0
1	0	-

$$A = W'_6 X'_6$$

X \ W	0	1
0	0	1
1	0	-

$$B = W_6 X'_6$$

$$C = 0$$

$$D = 0$$

VIII. DECREMENT LOGIC

A. Seconds Ones (0-9)

Z	Y	X	W	D	C	B	A
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

ZY \ XW	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	-	-	-	-
10	1	0	-	-

$$A = W'_1$$

ZY \ XW	00	01	11	10
00	0	0	1	0
01	1	0	1	0
11	-	-	-	-
10	1	0	-	-

$$B = (X'_1 W'_1 ((Z'_1 Y_1) + (Z_1 Y'_1))) + (Z'_1 W_1 X_1)$$

ZY	XW			
	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	-	-	-	-
10	1	0	-	-

ZY	XW			
	00	01	11	10
00	1	0	0	1
01	1	0	-	-
11	-	-	-	-
10	-	-	-	-

$$A = W_2'$$

$$C = (Z_1'Y_1(X_1 + W_1)) + (Z_1X_1'W_1'Y_1')$$

ZY	XW			
	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	-	-	-	-
10	0	1	-	-

ZY	XW			
	00	01	11	10
00	0	0	1	0
01	1	0	-	-
11	-	-	-	-
10	-	-	-	-

$$B = (Y_2X_2'W_2') + (Y_2'X_2W_2)$$

$$D = X_1'Y_1'((Z_1W_1) + (Z_1'W_1'))$$

ZY	XW			
	00	01	11	10
00	1	0	0	0
01	0	1	-	-
11	-	-	-	-
10	-	-	-	-

$$C = X_2'((Y_2W_2) + (Y_2'W_2'))$$

$$D = 0$$

C. Minutes Ones (0-9)

Same as Seconds Ones with W3/X3/Y3/Z3.

B. Seconds Tens (0-5)

Z	Y	X	W	D	C	B	A
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	1	0	0	1	0
0	1	0	0	0	0	1	1
0	1	0	1	0	1	0	0

D. Minutes Tens (0-5)

Same as Seconds Tens with W4/X4/Y4/Z4.

E. Hours Ones

I. Tens = 0/1 → 0-9

Same as Seconds Ones with W5/X5/Y5/Z5.

II. Tens = 2 → 0-3

X	W	D	C	B	A
0	0	0	0	1	1
0	1	0	0	0	0
1	0	0	0	0	1
1	1	0	0	1	0

X \ W	0	1
0	1	0
1	1	0

$$A = W'_5$$

X \ W	0	1
0	1	0
1	0	1

$$B = (X_5 W_5) + (X'_5 W'_5)$$

$$C = 0$$

$$D = 0$$

F. Hours Tens (0-2)

X	W	D	C	B	A
0	0	0	0	1	0
0	1	0	0	0	0
1	0	0	0	0	1

X \ W	0	1
0	0	0
1	1	-

$$A = X_6 W'_6$$

X \ W	0	1
0	1	0
1	0	-

$$B = X'_6 W'_6$$

$$C = 0$$

$$D = 0$$

IX. 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.

X. 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)

XI. EXECUTION

A. Upload Code to Arduino

- 1) Connect Arduino to computer via USB
- 2) Upload the following code to the Arduino using PlatformIO.

`https://github.com/gadepall/clock/blob/main/codes/code.cpp`
- 3) Open PlatformIO, select New Project and then fill in the details (name, board & framework).
- 4) Then replace contents in src/main.cpp with the above code, now run & upload that code to Arduino Uno.

B. Hardware Build

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

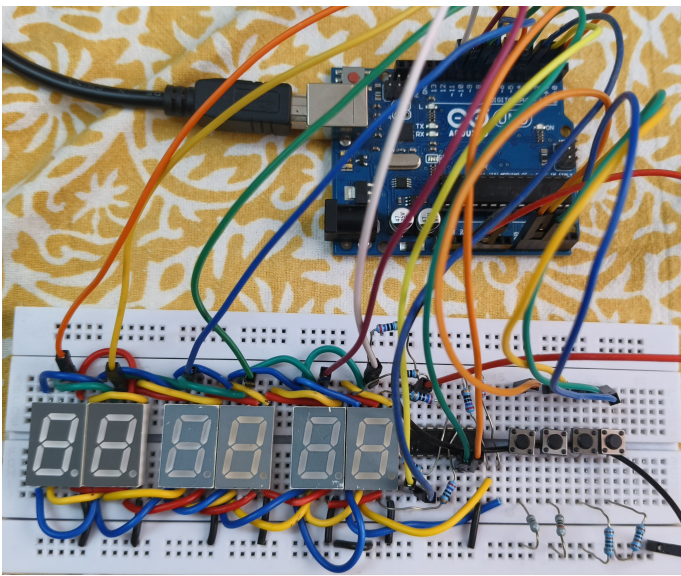


Fig. 1. Final Arduino-based Clock Implementation

FUTURE SCOPE

- Integration with wireless modules (Bluetooth/Wi-Fi) for remote time setting and synchronization.
- Addition of alarms, timers, and countdown features with user-defined events.
- Implementation of a real-time clock (RTC) module for improved accuracy and power efficiency.
- Expansion to a multi-language or multi-format (12/24-hour) display interface.
- Incorporation of IoT functionality for smart home or wearable applications.

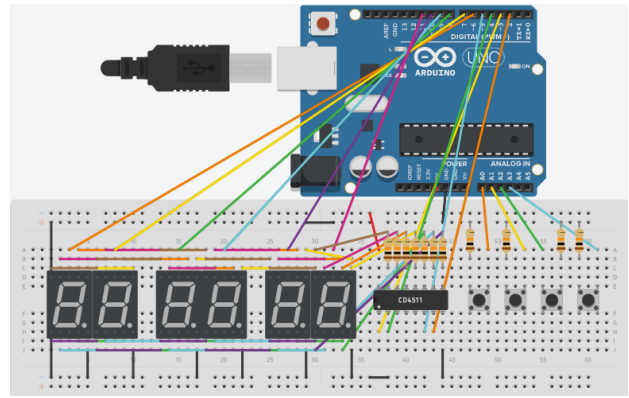


Fig. 2. Tinkercad Simulation of the Digital Clock

REFERENCES

- 1) M. Morris Mano, *Digital Design*, 5th ed., Pearson, 2013.
- 2) A. Malvino and D. Leach, *Digital Principles and Applications*, 8th ed., McGraw-Hill, 2017.
- 3) D. A. Patterson and J. L. Hennessy, *Computer Organization and Design*, 5th ed., Morgan Kaufmann, 2014.
- 4) Arduino Documentation, <https://www.arduino.cc/reference/en/>.
- 5) Datasheet for IC 7447, Texas Instruments, <https://www.ti.com/product/7447>.