Digital Clock using the Arduino Framework

Dhawal Saini and G. V. V. Sharma
Department of Electrical Engineering
Indian Institute of Technology Hyderabad
Email: gadepall@ee.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]–[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, a simple state machine representing a decade counter is implemented in [6]. Based on this, we design 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. Clock Functionality and Hardware Setup

Fig. 1 shows the various hardware connections and the corresponding components are listed in Table II. The clock follows the 23:59:59 format using 6 displays. The first 2 represent hours, next 2 denote minutes and the last 2 count the seconds. The six displays are connected to the arduino through the pin connections listed in Table I. The number count for each display is also listed therein. Pin connections from the Arduino to the four push buttons are available in Table III. Table IV gives the connections between the Arduino and the IC 7447 display decoder. . Further, the pin connections for the IC 7447 to all the displays are available in Table V.

Display	Arduino Pin	Description
1	D4	Counts 0-2
2	D5	Counts $\begin{cases} 0-3 & display1 = 2, \\ 0-9 & otherwise \end{cases}$
3	D6	Counts 0-5
4	D7	Counts 0-9
5	D8	Counts 0-5
6	D9	Counts 0-9

TABLE I
Display to Arduino Connections

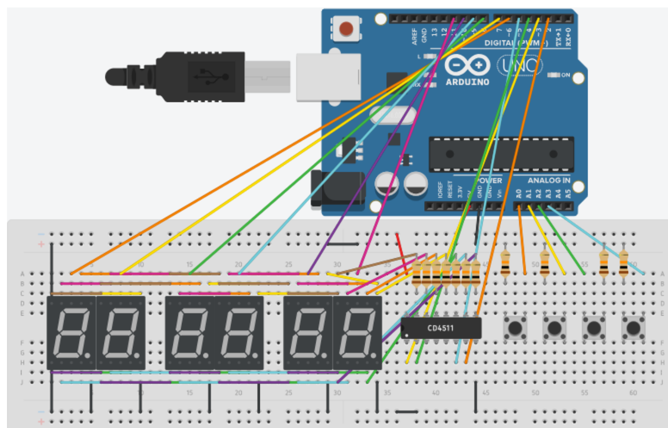

Fig. 1. Tinkercad Simulation of the Digital Clock

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 II
Components List

Button	Arduino Pin	Description
1	D10	Edit Mode Toggle
2	D11	Next Digit Selection
3	D12	Increment Digit
4	D13	Decrement Digit

TABLE III
Button to Arduino Connections

III. Clock Logic

A. Displays 2 (Display1 $\neq 2$), 4 and 6

The truth table is available in Table VI and the corresponding K-maps using dont cares for the output variables are given

IC 7447 PIN	Arduino Pin	Description
7	D0	BCD Bit 0 (A)
1	D1	BCD Bit 1 (B)
2	D2	BCD Bit 2 (C)
6	D3	BCD Bit 3 (D)

TABLE IV
IC 7447 TO ARDUINO CONNECTIONS

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

TABLE V
BCD TO 7-SEGMENT CONNECTIONS

in Figs. 2-5 yielding the following logic equations.

$$A = W_1' \tag{1}$$
$$B = (W_1 X_1' Z_1') + (W_1' X_1) \tag{2}$$
$$C = (X_1' Y_1) + (W_1' Y_1) + (W_1 X_1 Y_1') \tag{3}$$
$$D = (W_1' Z_1) + (W_1 X_1 Y_1) \tag{4}$$

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

TABLE VI

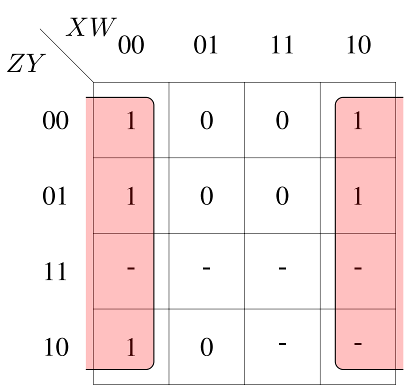

Fig. 2. A

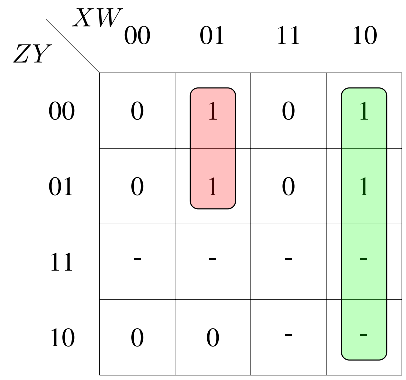

Fig. 3. B

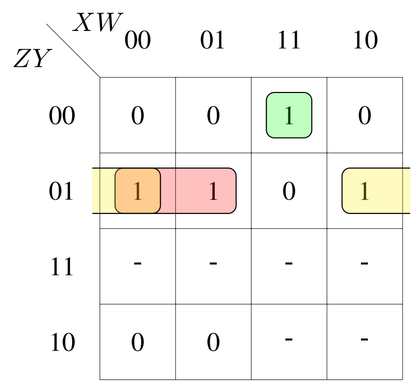

Fig. 4. C

B. Displays 3 and 5

The truth table is available in Table VII and the corresponding K-maps using dont cares for the output variables are given in Figs. 6-8 yielding the following logic equations.

$$A = W_2' \tag{5}$$
$$B = (W_2 X_2' Y_2') + (W_2' X_2) \tag{6}$$
$$C = (W_2 X_2) + (W_2' X_2' Y_2) \tag{7}$$
$$D = 0 \tag{8}$$
$$\tag{9}$$

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	0	0	0

TABLE VII

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

Fig. 5. D

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

Fig. 6. A

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

Fig. 7. B

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

Fig. 8. C

C. *Display 2 (Display1 = 2)*

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

$X \backslash W$	0	1
0	1	0
1	1	0

$$A = W_5'$$

$X \backslash W$	0	1
0	0	1
1	1	0

$$B = (W_5 X_5') + (W_5' X_5)$$

$$C = 0$$
$$D = 0$$

D. *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

$X \backslash W$	0	1
0	1	0
1	0	-

$$A = W_6' X_6'$$

$X \backslash W$	0	1
0	0	1
1	0	-

$$B = W_6 X_6'$$

$$C = 0$$
$$D = 0$$

IV. 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

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

$$A = W_1'$$

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

$$B = (X_1' W_1' ((Z_1' Y_1) + (Z_1 Y_1'))) + (Z_1' W_1 X_1)$$

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

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

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

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

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

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	0	1	0
01	1	0	-	-
11	-	-	-	-
10	-	-	-	-

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

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.

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	-

$$B = X_6' W_6'$$

$$C = 0$$
$$D = 0$$

V. 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.

VI. 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.

VII. 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.

VIII. 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)

IX. 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

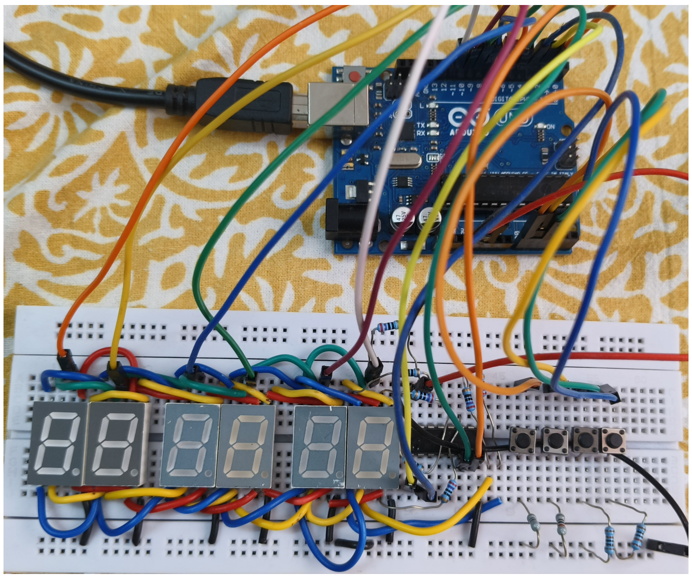

Fig. 9. Final Arduino-based Clock Implementation

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.

References

[1] M. M. 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, *Arduino Reference*, 2025, accessed: October 21, 2025. [Online]. Available: https://www.arduino.cc/reference/en/
[5] Texas Instruments, *Datasheet for IC 7447*, 2025, accessed: October 21, 2025. [Online]. Available: https://www.ti.com/product/7447
[6] G. V. V. Sharma, *Digital Design through Arduino*, 3rd ed. Narayanpal Prakashan, 2025.