Digital Clock using the Arduino Framework

Dhawal Saini
Department of Electrical Engineering
Indian Institute of Technology Hyderabad

G. V. V. Sharma
Department of Electrical Engineering
Indian Institute of Technology Hyderabad
gadepall@ee.iith.ac.in

January 29, 2026

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.

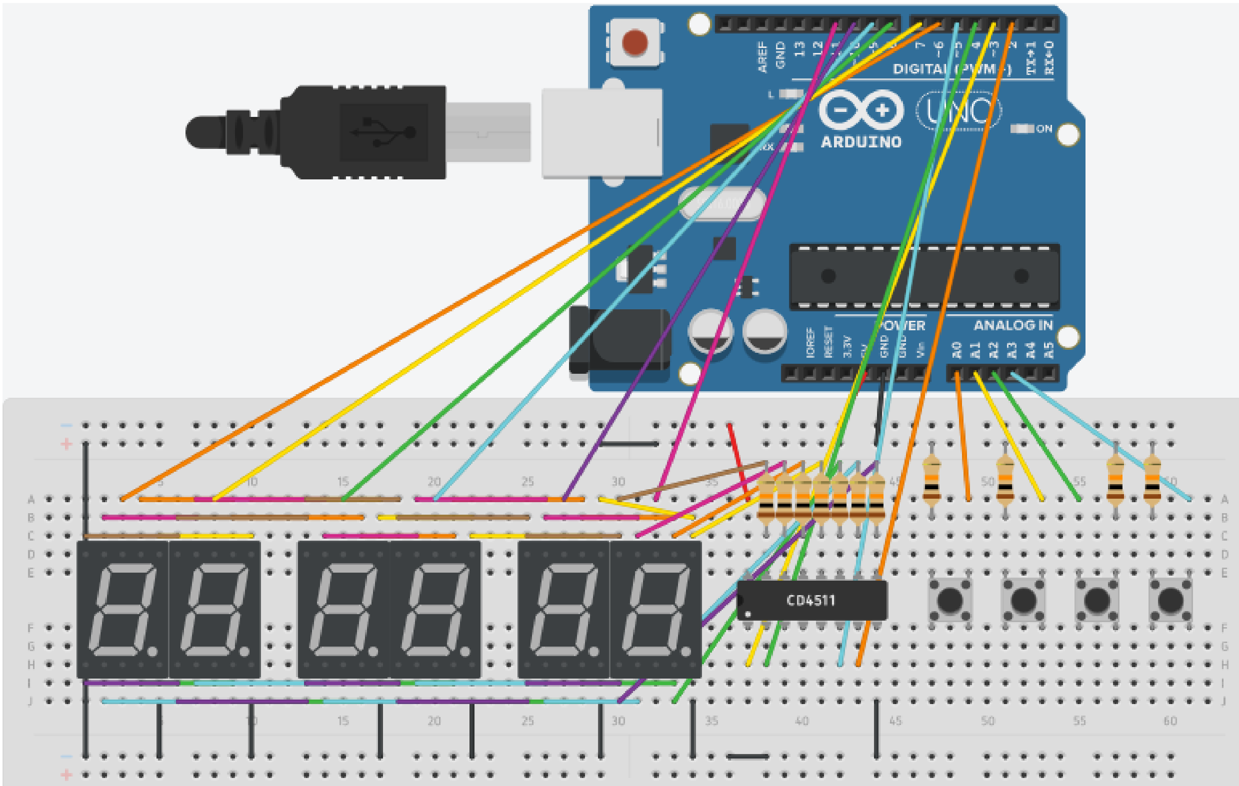


Figure 1: Tinkercad Simulation of the Digital Clock

1 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 (Mano, 2013; Malvino and Leach, 2017; Patterson and Hennessy, 2014). The advent of microcontroller platforms, particularly Arduino, has enabled the development of compact, programmable clocks with enhanced user interactivity (Arduino Documentation, 2025). Techniques such as BCD-to-seven-segment interfacing and display multiplexing allow efficient utilization of limited I/O resources while maintaining accurate visual representation (Texas Instruments, 2025). Inspired by these principles, a simple state machine representing a decade counter is implemented in (Sharma, 2025). 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. Most digital clock designs rely on standard counters as the building blocks. However, our design is simpler since it relies on the state machine in (Sharma, 2025).

2 Clock Functionality and Hardware Setup

Fig. 1 shows the various hardware connections and the corresponding components are listed in Table 2. The clock follows the 23:59:59 format using six displays. The first two represent hours, the next two denote minutes, and the last two count the seconds. The six displays are connected to the Arduino through the pin connections listed in Table 1. The number count for each display is also listed therein.

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 1: Display to Arduino Connections

Pin connections from the Arduino to the four push buttons are available in Table 3. Table 4 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 5.

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 2: 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 3: Button to Arduino Connections

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 4: 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 5: BCD to 7-Segment Connections

3 Clock Logic

Here, we discuss how to design counters for the various displays in the clock.

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

These displays count from 0-9. Display 2 will do this only when Display 1 is less than 2. The truth table and the corresponding K-maps using dont cares for the output variables are given in Fig. 2 along with the corresponding logic equations.

3.2 Displays 3 and 5

These displays count from 0-5. The truth table and the corresponding K-maps using dont cares for the output variables are given in Fig. 3, along with the corresponding logic equations. $D = 0$ in this case.

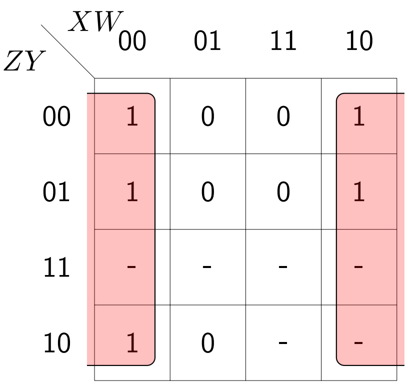
3.3 Display 2 (Display1 = 2)

This display counts from 0-3, when the first display shows 2. The truth table and the corresponding K-maps using dont cares for the output variables are given in Fig. 4. along with the corresponding logic equations. $C = D = 0$ in this case.

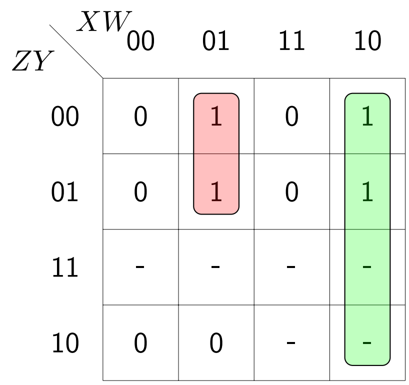
3.4 Display 1

This display counts from 0-2, representing the first digit of the hour. The truth table and the corresponding K-maps using dont cares for the output variables are given in Fig. 5, along with the corresponding logic equations. $C = D = 0$ in this case.

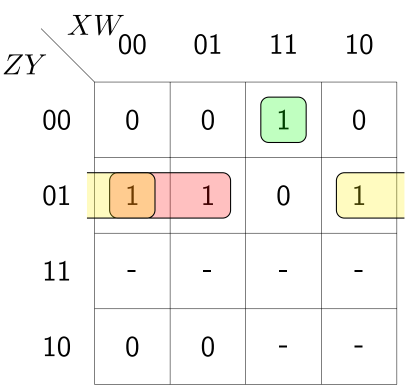
Z	Y	X	W	D	C	B	A
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0



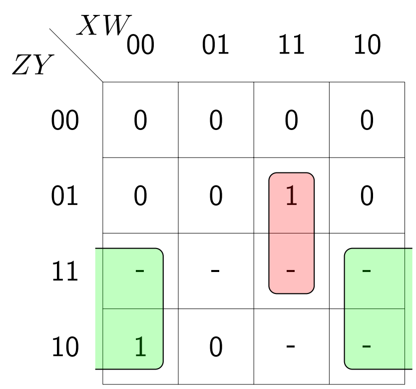
(a) $A = W_1'$



(b) $B = W_1X_1'Z_1' + W_1'X_1$



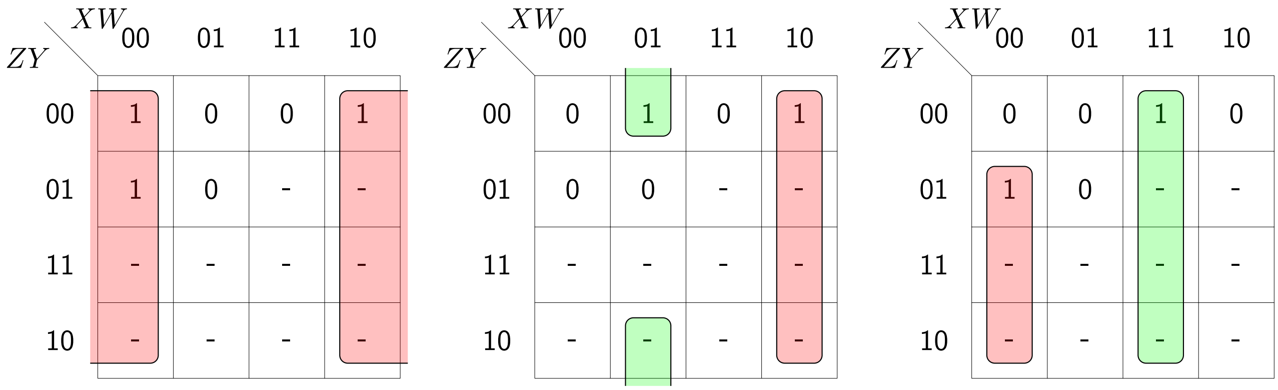
(c) $C = X_1'Y_1 + W_1'Y_1 + W_1X_1Y_1'$



(d) $D = W_1'Z_1 + W_1X_1Y_1$

Figure 2: Counting 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	0	0	0



(a) $A = W_2'$

(b) $B = W_2X_2'Y_2' + W_2'X_2$

(c) $C = W_2X_2 + W_2'X_2'Y_2$

Figure 3: Counting 0-5

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

(a) $A = W_5'$

(b) $B = W_5X_5' + W_5'X_5$

Figure 4: Counting 0-3

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

(a) $A = W_6' X_6'$

(b) $B = W_6 X_6'$

Figure 5: Counting 0–2

4 Control Logic and Implementation

4.1 Decrement Logic

This logic is primarily used to decrease the count on a particular display using button 4. This is helpful in setting the time for the clock. The relevant truth tables and resulting Boolean equations using K-maps following the approach in the previous section are available in Table 6.

4.2 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 (see Table 1). 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.

4.3 Digit Editing

The clock allows pausing and digit-by-digit editing (See Table 3).

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.

4.4 Implementation

Pressing Button 1 toggles between run mode and edit mode. In edit mode, the clock pauses. Following functions are available in edit mode. The selected digit blinks at 5Hz (200ms on, 200ms off) for visual feedback.

1. Pressing Button 2 selects the next digit for editing (cycles through all six digits).

2. Pressing Button 3 increments the currently selected digit using the increment logic tables.
3. Pressing Button 4 decrements the currently selected digit using the decrement logic tables.

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

$$A = W_1' \tag{1}$$
$$B = (X_1'W_1'((Z_1'Y_1) + (Z_1Y_1'))) + (Z_1'W_1X_1) \tag{2}$$
$$C = (Z_1'Y_1(X_1 + W_1)) + (Z_1X_1'W_1'Y_1') \tag{3}$$
$$D = X_1'Y_1'((Z_1W_1) + (Z_1'W_1')) \tag{4}$$

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

$$A = W_2' \tag{5}$$
$$B = (Y_2X_2'W_2') + (Y_2'X_2W_2) \tag{6}$$
$$C = X_2'((Y_2W_2) + (Y_2'W_2')) \tag{7}$$
$$D = 0 \tag{8}$$

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

$$A = W_5' \tag{9}$$
$$B = (X_5W_5) + (X_5'W_5') \tag{10}$$
$$C = D = 0 \tag{11}$$

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

$$A = X_6W_6' \tag{12}$$
$$B = X_6'W_6' \tag{13}$$
$$C = D = 0 \tag{14}$$

Table 6: Truth Tables for decrementing displays

5 Software

Following are the steps to upload code to the Arduino

1. Connect Arduino to computer via USB
2. Upload the following code to the Arduino

https://github.com/gadepall/clock/blob/main/codes/code.cpp

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)

6 Future Work

The clock prototype on the Arduino is the first step towards building a chip. The next step is to adapt the current design for implementation on a Vaman FPGA with a QuickLogic SoC and use additional EDA tools for physical chip design. Many real time issues like flicker or timing drift will also be considered in the process and will be addressed in future work.

References

Arduino Documentation. *Arduino Reference*, 2025. URL https://www.arduino.cc/reference/en/. Accessed: October 21, 2025.

A. Malvino and D. Leach. *Digital Principles and Applications*. McGraw-Hill, 8th edition, 2017.

M. Morris Mano. *Digital Design*. Pearson, 5th edition, 2013.

D. A. Patterson and J. L. Hennessy. *Computer Organization and Design*. Morgan Kaufmann, 5th edition, 2014.

G. V. V. Sharma. *Digital Design through Arduino*. Narayanpal Prakashan, 3rd edition, 2025.

Texas Instruments. *Datasheet for IC 7447*, 2025. URL https://www.ti.com/product/7447. Accessed: October 21, 2025.