

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

Electrical Engineering Department
Indian Institute of Technology, Hyderabad

Outline

- 1 Introduction
- 2 Components
- 3 Circuit Connections
- 4 Multiplexing Technique
- 5 Digit Editing Logic
- 6 Conclusion

Introduction

- Digital clock system with editing capabilities using Arduino microcontroller
- Key features:
 - Multiplexing technique for six 7-segment displays
 - Minimal I/O pin usage
 - Pause/play functionality
 - Digit-by-digit editing with increment/decrement buttons
- Comprehensive Boolean logic for time constraints

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	4

Table: Components List

Arduino Pin Connections

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

Multiplexing Technique

- All segment inputs connected to single BCD decoder
- Digital pins control common cathode of each display
- Selective activation of displays
- 2ms time gap between display switching
- Creates illusion of simultaneous operation
- Minimal I/O pin usage (only 6 pins for displays)

Editing System Overview

- ① Button 1: Toggles between run mode and edit mode
- ② Button 2: Cycles through six digits in edit mode
- ③ Button 3: Increments selected digit with rollover constraints
- ④ Button 4: Decrements selected digit with rollunder constraints

Different constraints based on digit position:

- Units digits: 0-9
- Tens of minutes/seconds: 0-5
- Tens of hours: 0-2

Increment Logic - 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

Simplified Boolean expressions:

$$A_1 = A'$$

$$B_1 = A + B$$

$$C_1 = AB + C$$

$$D_1 = ABC + D$$

Increment Logic - 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:

$$A_1 = A'B'C' + AB'C$$

$$B_1 = A'BC' + AB'C'$$

$$C_1 = A'BC + AB'C$$

$$D_1 = 0$$

Increment Logic - Tens of Hours (0-2)

Simplified Boolean expressions:

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

$$A_1 = A'B'$$

$$B_1 = AB'$$

$$C_1 = 0$$

$$D_1 = 0$$

Decrement Logic - 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

Simplified Boolean expressions:

$$A_1 = A$$

$$B_1 = A' + B$$

$$C_1 = A'B' + C$$

$$D_1 = A'B'C' + D$$

Decrement Logic - 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:

$$A_1 = A'B'C + AB'C'$$

$$B_1 = A'BC' + AB'C'$$

$$C_1 = A'B'C' + ABC'$$

$$D_1 = 0$$

Decrement Logic - Tens of Hours (0-2)

Simplified Boolean expressions:

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

$$A_1 = BA'$$

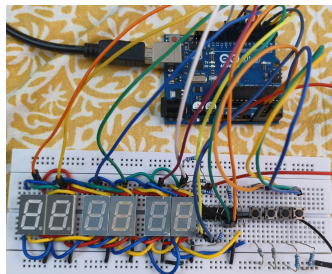
$$B_1 = B'A'$$

$$C_1 = 0$$

$$D_1 = 0$$

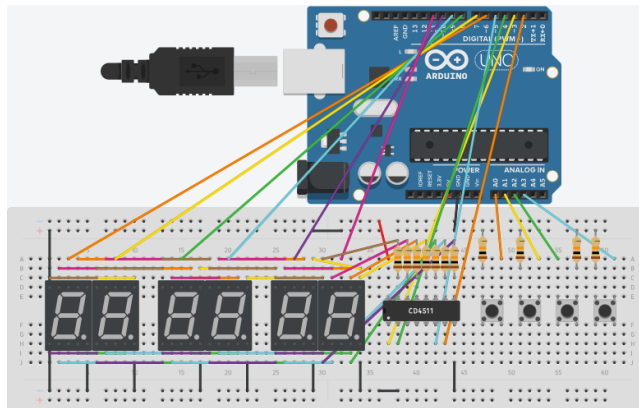
Hardware Build

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



Final Arduino-based Clock Implementation

Tinkercad Simulation



Digital Clock

Tinkercad Simulation of the

Summary

- Successfully implemented digital clock with editing features
- Key achievements:
 - Efficient multiplexing technique
 - Comprehensive editing system
 - Boolean logic implementation
 - Minimal I/O pin usage
- Complete documentation and source code available

Acknowledgment

The complete source code and documentation can be found at:

<https://github.com/Dhawal24112006/projects.git>

Thank You!