Hardware Assignment
Digital Clock

EE1003

Homa Harshitha Vuddanti
(EE24BTECH11062)

Problem Statement: Make a digital clock using seven-segment displays in AVR-GCC, without using flip-flops.

1 Introduction

I have used multiplexing for this project. A digital clock typically has multiple digits (e.g., hours, minutes, seconds) displayed on a 7-segment LED display. The microcontroller controlling the clock rapidly switches between the different digits, illuminating the segments of each digit in sequence.
The switching happens so quickly that the human eye perceives all digits lit simultaneously, even though only one digit is actually illuminated at any given moment. This technique significantly reduces the number of display pins required, as each digit doesn't need its own dedicated pins.

2 Materials Required

- ATmega328P (Arduino Uno)
- Breadboard
- Common Anode 7-Segment Displays
- Resistors (for segment control)
- Connecting Wires
- Push buttons (for manually setting time)

3 Procedure

1) **Hardware Setup:**
   - Connect the common-anode 7- segment displays to the ATmega328P via breadboard.
   - Assign segment control lines; I have used PD2-PD7, PB0 for segments a-g.
   - Assign common pins for digit selection: A0-A2 (PC0-PC2), 10-12 (PB2-PB4).
   - Connect push buttons to PB1, PB5, PC3-PC5 for input control.
   - We can actually use any ports, but we just have to make sure that the changes are made in the code as well.
2) **Software Implementation:** The code has following parts in it :

- the seg_map function is like a lookup table for checking digits display configuration.
- the set_segments function is used to send a number to the 7-segment display to correctly light up the segments.
- the enable_digit function is used to activate turn off all digits before enabling the correct one. We are basically mapping numbers to their ports.
- The ISR(TIMER1_COMPA_vect) function is to increment correctly. It increments each second, changes minutes when 60 seconds are crossed and so on.
- check_buttons function is for button control, if the variable set_mode is 1, it means we can use our assigned buttons to increment hours, minutes, seconds using them.
- The main function uses the above functions, configures pins, pulls up resistors, and 1-second interrupts. It has a multiplexing loop.

4 Button description

There are five buttons implemented here.

1) First one connected to PB1 is to stop or start time to set it. Let's call it the set button. When we first press this, clock stops ticking.
2) Second button connected to PB5 is to increment hours in the clock once we have stopped the clock using the set button.
3) the third and fourth buttons are used to set the minutes. One increments ten's of minutes and other the one's.
4) The last button connected to PC5 is to increment seconds.

5 Code

The code is available in the following git hub permalink :
https://github.com/HomaHarshitha/Digital-Clock--AVR-GCC/tree/44627885a86f58260135cd2e80b99632bd9f1763/Digital_clock-AVRGCC-ee062

6 Result

The digital clock successfully displays hours, minutes, and seconds using a common anode 7-segment display. Time can be adjusted using the provided push buttons. The multiplexing technique ensures clear and stable digit display.