COURSE PROJECT

EE1003

SCIENTIFIC PROGRAMMING

---

24 Hour Digital Clock

Contents

1 Introduction

This report documents the design, implementation, and functionality of a seven-segment clock built using the Arduino platform and AVR GCC toolchain.

The seven-segment clock combines hardware components including Arduino microcontroller, seven-segment displays, and supporting circuitry to create a digital clock that accurately displays hours, minutes, and seconds. The system was programmed using AVR GCC, allowing for direct control of the microcontroller's capabilities without relying on the standard Arduino IDE and libraries.

The following sections will detail the design considerations, component selection, circuit implementation, software architecture, challenges encountered during development, and potential future improvements for the system.

2 Hardware Implementation

TABLE 0: Components Used in Seven-Segment Clock Project

Component	**Quantity**	**Description**
Arduino Board	1	Microcontroller platform (Arduino Uno/Nano with ATmega328P)
Seven-Segment Displays	6	LED digital displays for showing time digits
BCD to Seven-Segment Decoder	1	IC for converting binary-coded decimal to seven-segment display format (e.g., CD4511 or 74HC47)
Push Buttons	3	Momentary tactile switches for setting time and controlling display modes
Resistors (220Ω)	7-8	Current-limiting resistors for seven-segment display segments
Resistors (10kΩ)	3	Pull-down/pull-up resistors for push buttons
Jumper Wires	Multiple	For connecting components on breadboard or PCB
Breadboard	1	For prototyping the circuit
Power Supply	1	5V power supply (via USB or external adapter)

The circuit diagram of the clock is given by fig. 0. There are six 7-segment displays(`DIGIT1 - DIGIT6`), three buttons (`S1 - S3`), one 7447 BCD to 7-Segment decoder, and one Arduino.

3 Software Implementation

The code for the driver:

```
./code/main.c
```

The above code runs the clock. Like said before, we use the concept fo multiplexing. By that, I mean, I loop through each 7-segment display and power them for a short enough time (2ms). This will make it look like all the six of them are powered at the same time.

I have made three separate functions that increment hours, minutes and seconds separately and a nested if-else to join them all together.

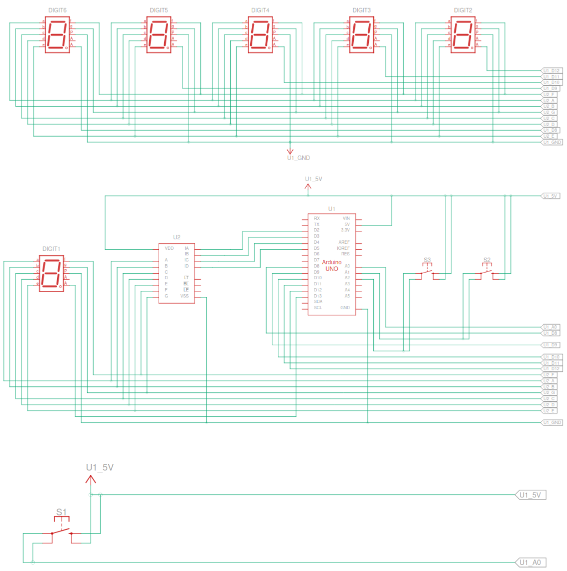

Fig. 0: Circuit diagram

4 User Interface

The first mode is normal function mode, where it displays currently set time(`normal mode`). In mode two, we can set time being displayed(`set mode`).To switch between the modes, we press `S1`.

Once you enter the `set mode`, the seconds segment will start blinking. This indicates currently seconds is selected to change. To circulate between seconds, minutes and hours segments, press `S`. The currently selected segment will blink. To change the time displayed the current segment, press `S3`. For hours and minutes, the time increments, meanwhile for seconds, it resets back to `00`. The clock doesn't stop ticking in this mode.

In `normal mode`, the clock runs normally and buttons `S2` and `S3` are disabled.

5 Conclusion

There are multiple improvements that can be done in the clock. First of all, the clock is not very accurate and looses a litte bit of time every day. Next thing that could be done is add an alarm system, making it further useful. As it is more of a clock and not a watch, we wouldn't need a stopwatch or a timer.

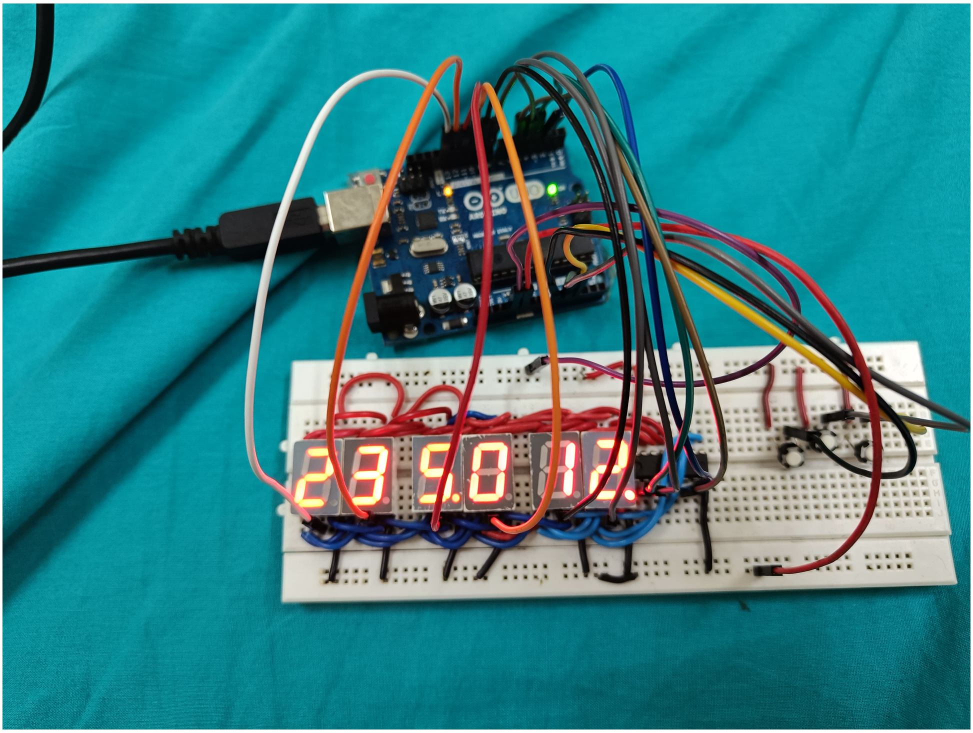

Fig. 0: Physical Build