TOSHKENT SHAHRIDAGI INHA UNIVERSITETI
INHA UNIVERSITY IN TASHKENT

Embedded Software & Design (SOC3050)

Requirement Analysis Document

Digital Alarm Clock by Team IMPERIUM

Team members (Name, ID, Group/Section):

1. Oybek Amonov	U1610176	CSE16-2/001
2. Rakhmatjon Khasanov	U1610183	CSE16-2/001
3. Bokhodir Urinboev(leader)	U1610249	CSE16-1/001

Table of Content

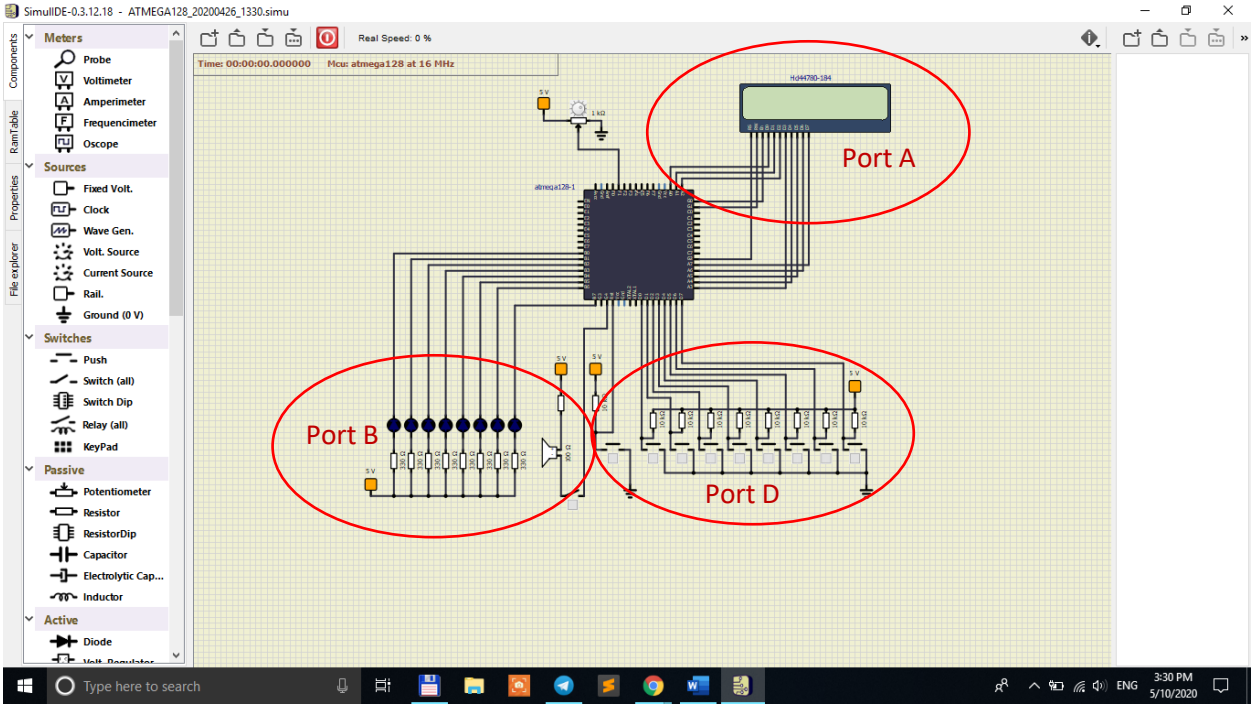
Introduction 3
Requirement Analysis 3
Conclusion 7
References 7

Introduction

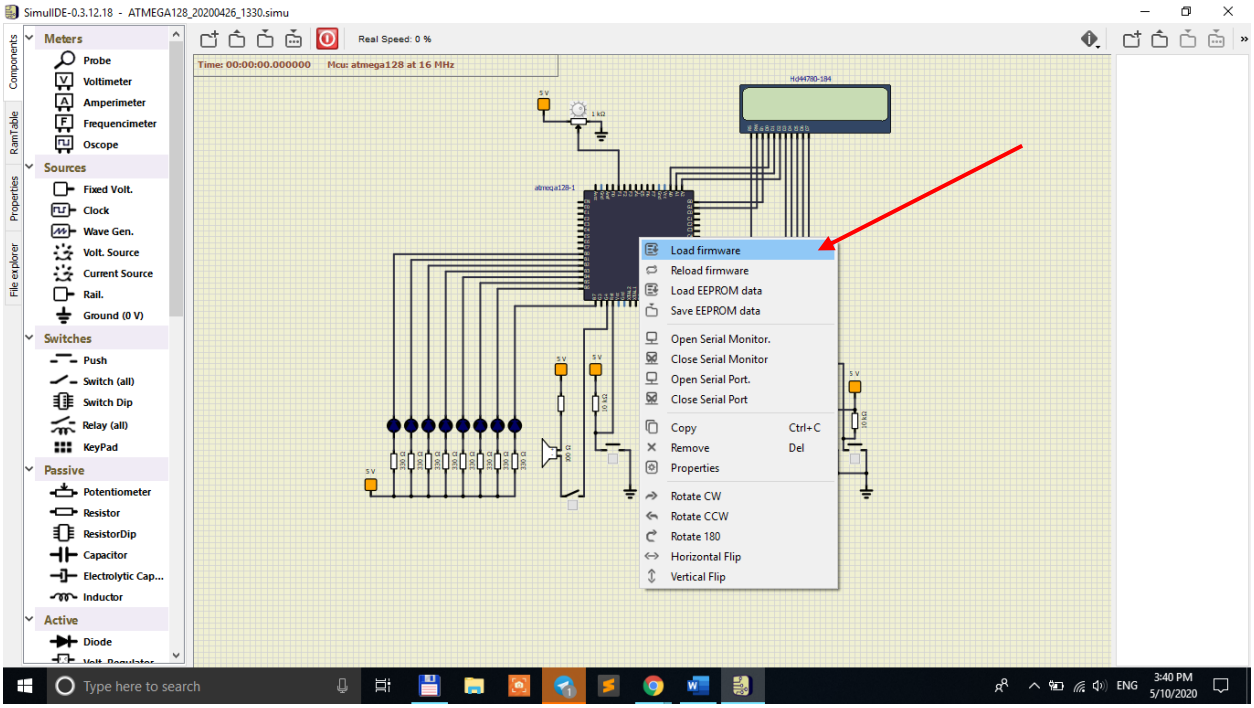
This document is to provide thorough understanding of requirement list for Digital Clock using Atmega128 microcontroller. This project is intended to provide students with basic concepts about working with microcontrollers. Students are supposed to learn how to use LEDs, LCDs, and Ports of the microcontroller Atmega128. By the end of the project, Digital Clock must have following functions: Mode selection (Time Display, Alarm Clock, Stopwatch, Set Time/Alarm). Prior Timer and Clock knowledge in microcontrollers, and hardware specification knowledge is welcomed before reading the document.

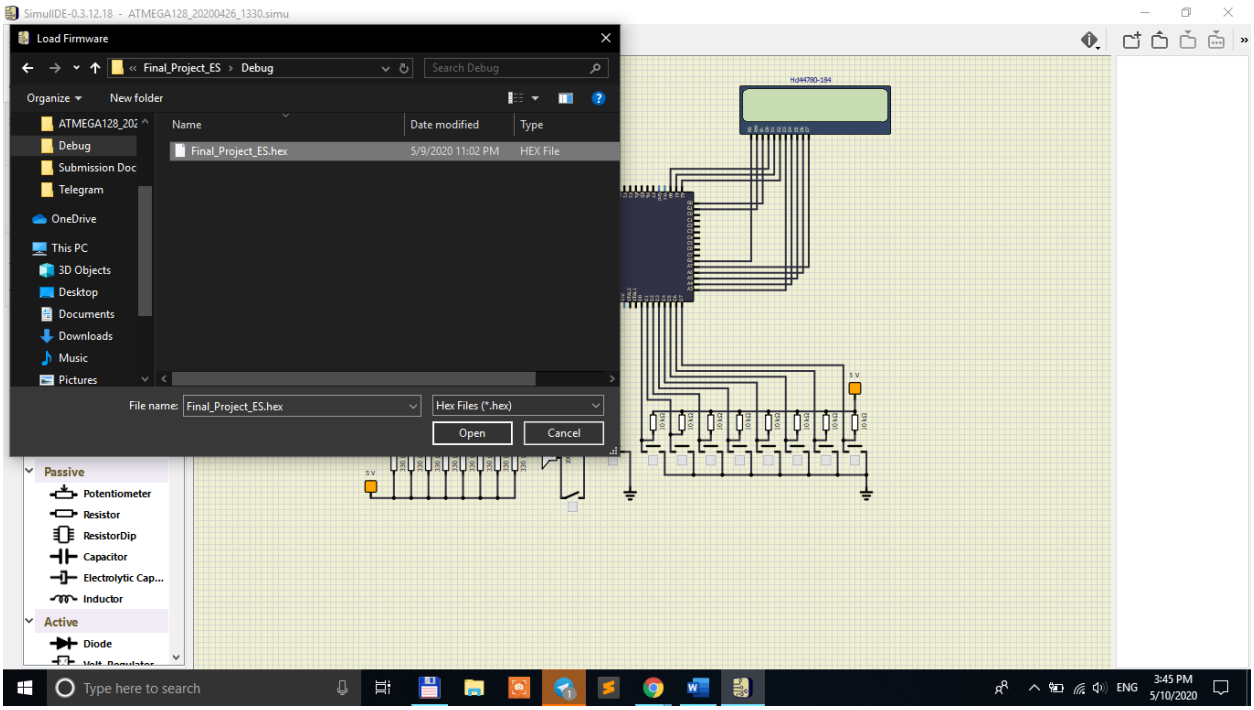
Requirement Analysis

The clock must be implemented using either assembly or C language. It must be run on SimulIDE to show the project correctness and goal achievements. It is known that microcontrollers work without CPUs, and they run only machine code embedded in them. So, all code written in assembly/C languages must be interpreted into machine code (.hex file in our case). When the project is run on the simulator it must show the Menu part of the project, so the user will choose the intended mode. The following picture shows how atmega128 microcontroller circuit is implemented. As you can see it has external trigger port (Port D) to give some input to the microcontroller, also there is a LED connected to Atmega128 through Port B. Last but not least, LCD display is connected the controller through Port A.

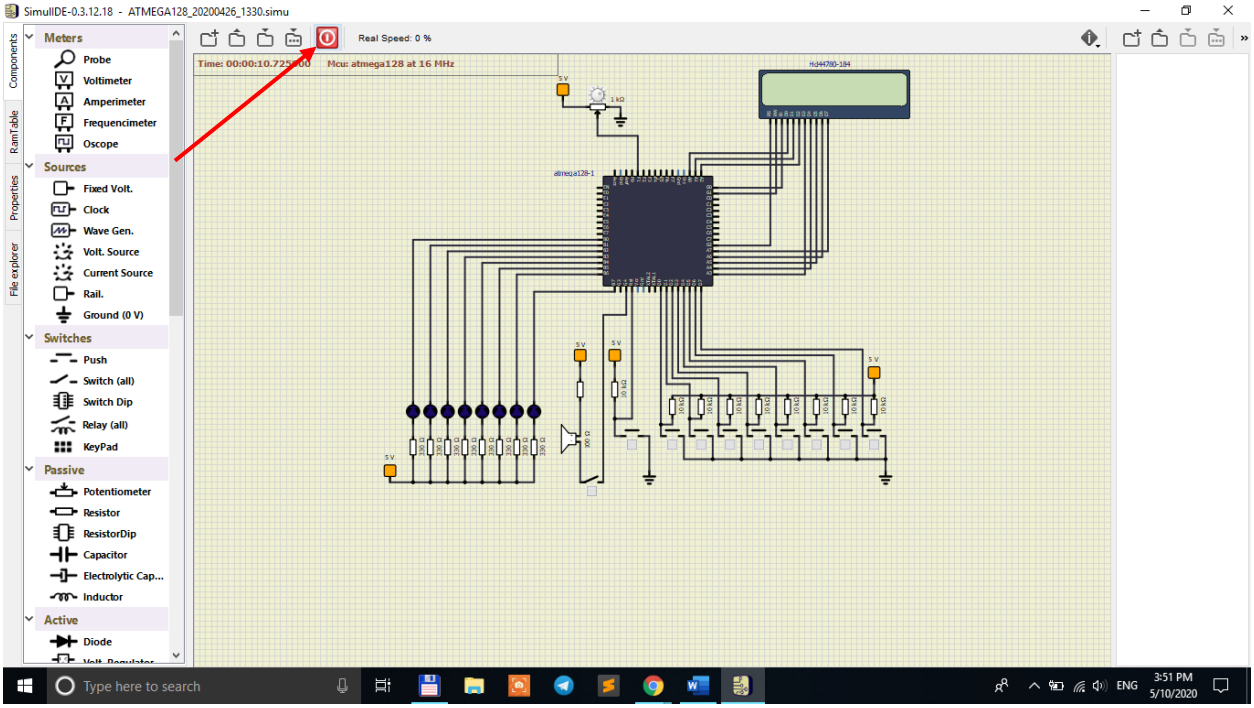


The software must be loaded using .hex file built using Atmel Studio. You should right-click on the microcontroller and choose “Load firmware” option. And choose the appropraite .hex file.

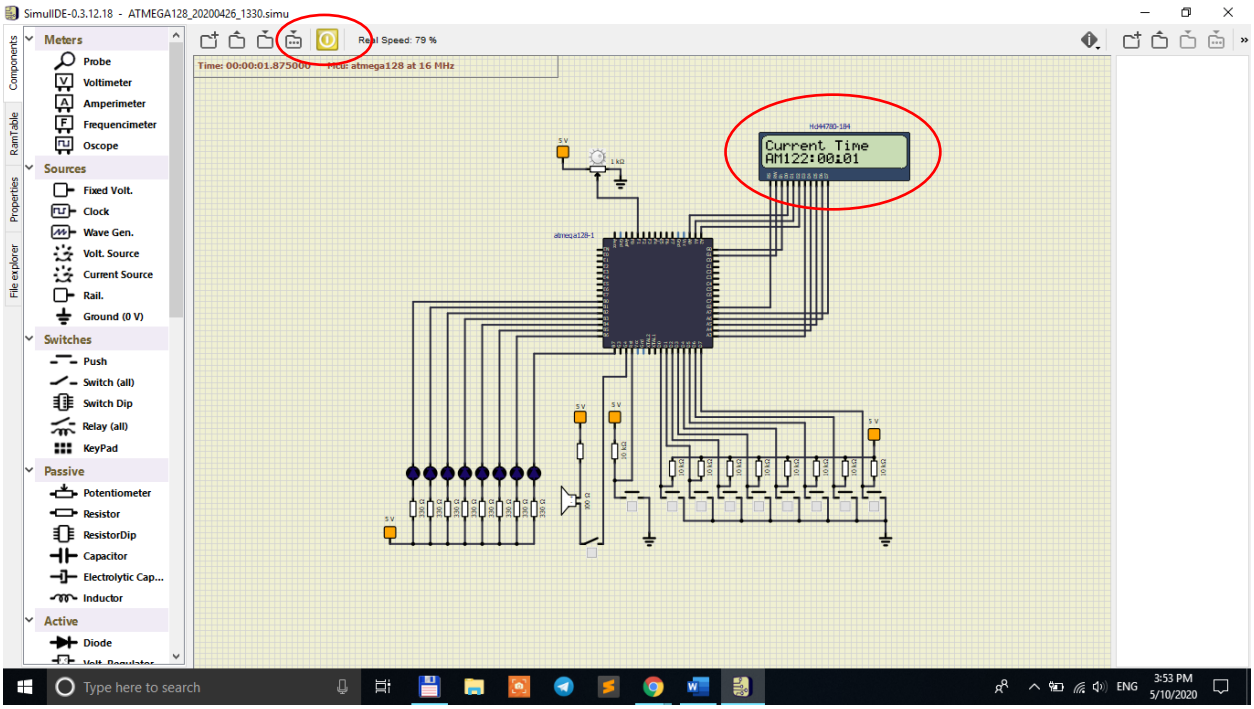




After successfully loading the .hex file, we can turn on the circuit and use the software.

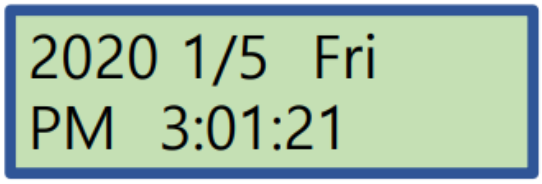


The following is the run-time screenshot of time_displsy.hex file downloaded from eclass.inha.uz. Our project must also have a Display Time Mode where current time is shown using LCD.

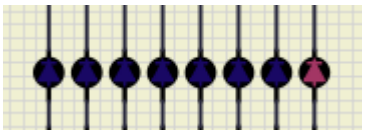


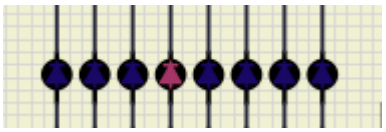
However, our display mode must be a bit different from above example, it should be as following:

- Line1 must have YYYY DD/MM WEEKDAY format
- Line2 must have AM (or PM) HH:MM:SS format

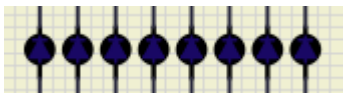


All information must be set using Set time/alarm modes, otherwise default values will be used to display the time. Next Mode is about Alarm mode, where we have to set alarm and when it is time for alarming, we need to use LED blinking to show that our Alarm Time is set correctly. The following pictures show that our led is blinking one at a time and in right

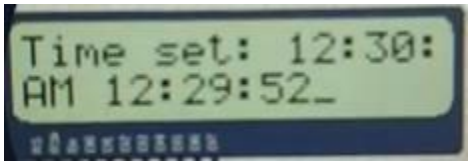




direction, but our project will have to blink them all at once as following:



What's more, we plan to show the alarm time on LCD too. Possible output is intended to be likewise:

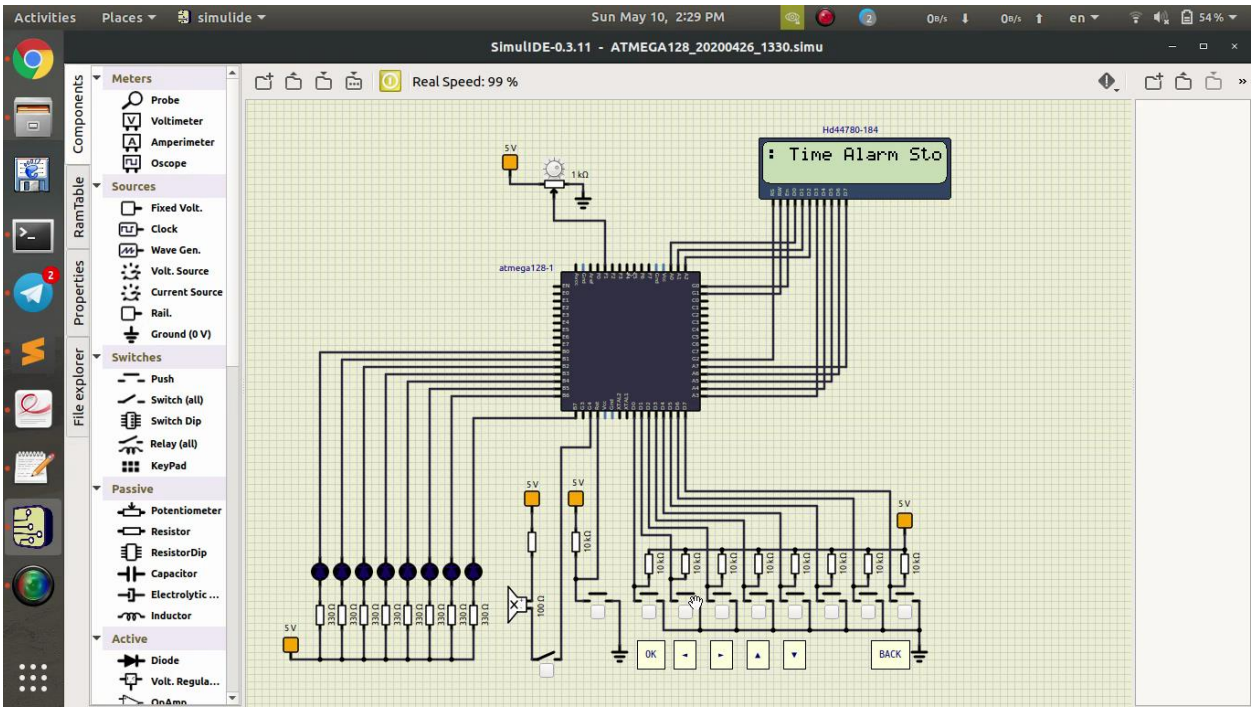


As you see from the figure that alarm is set to 12:30, and it is 8 seconds left till the alarm time. This result is achieved during implementation phase so it might change by the end of project. But main concept is based on that. The last mode is Stopwatch mode, which needs to have Start/Pause/Reset options. And the output mode is quite obvious, the format must be as following:

HH:MM:SS.mm (using 1/100 resolution for milliseconds)

Conclusion

The requirements for Digital Clock with Atmega128 microcontroller are the mentioned above, and they must be implemented by correct prescaling of clock on Atmega. The period of microcontroller is 14.7456 MHz (but we are using SimulIDE atmega128 circuit which has 16 MHz of clock cycle), and Atmega allows prescaling of maximum clock/1024. Thus, we can modify the clock ticking to almost to 1 second. Whole program will be implemented using external interrupts to select and change between modes, and timer interrupts are allowed to get interrupts for time handling. Proper functions must be implemented prior to implementing the main program, they include LED on/off, LED blinking, LED shifting, LCD initialization, LCD clear, LCD positioning, LCD output functions. As mentioned earlier, Port D is used for external interrupts. Possible division of buttons is as following, to select the mode user will click on button2(int1) and button3(int2) to navigate through. Button1(int0) to select mode, and button4(int3) to go back. On LCD, there is a displayed menu and it is probably shifted left continuously. The selected mode is Blinking on LCD, so by clicking OK button we can change the mode of the software. These features are subject to change during the implementation phase.



References

[1] https://www.youtube.com/watch?v=PNlSHrDM3jo – Sample Digital Clock demo video

[2] https://www.patreon.com/posts/simulide-0-3-12-35657927 - SimulIDE

[3] https://drive.google.com/open?id=1vBCQeEhz8_Ln2E0ZIGyQyJ3Yi8BxVU9q – Atmega128 Circuit