EEEN202 2021 Lab 1, Part C

Assembler, counter – timers and interrupts

During this lab session, the idea is to become acquainted with implementing interrupts. The task is again to write some code to implement a digital clock, but this time a timer interrupt is used instead of polling.

Create a project. From the menu, click `Project -> New uVision Project`. Name this project `Clock2`, and set the Project location to a folder where you want to store the project.

KEIL will then ask you to select your target Device.

- Device – Atmel – AT89C51AC3
- Click OK.

Next it will ask you if you want to copy a version of 8051 Startup code. Select **“No”**

Add “**clock2.a51**” to the group and double click the file to open it for editing.

Now open file “Clock2lab.a51” and copy the text into “clock2.a51”. Save “clock2.a51”, this code is to give you a head start in building your clock.

You will need to finish the setup code to enable the interrupt as well writing the Interrupt Service Routine (ISR).

Remember, before compiling your project, you must configure the flash tools. Look at the last lab for details.

Once you have made the necessary changes you can compile and download your program.

Press the RESET button to run the program. **You must press reset after each new upload to run your program.** Verify that your clock is working.

Lab 1 write up: The purpose of these labs is to introduce you to "low" level programming. This approach was common in the past for embedded systems but today has largely been replaced with higher level language approaches such as "C". For your write up you should cover the following:

1. Discuss the operation of the timer unit and we use it to make our one second interval.
2. Give an overview of the code for both the polling and interrupt implementations of your clock. Then discuss the pros and cons of the two methods.
3. Discuss the pros and cons of the "low" level programming approach versus "high" level language approaches.

Additional questions to be answered as part of your lab report:

For the 8051 Microcontroller:

1. Write an instruction sequence that could be used to read bit 1 of Port 0.

2. What addressing mode is used to access the upper 128 bytes of internal RAM?

3. Show how the content of internal address 6BH could be transferred to the accumulator.

4. What is the difference between the following instructions: ADD A,@R5 and ADD A,R5

5. Below shows a sequence of instructions, give the result of accumulator before and after the DA instruction.

   ```
   MOV   A,#13H
   MOV   R2,#18H
   ADD   A, R2
   DA    A
   ```

6. Explain the difference between AJMP, SJMP, and LJMP instructions.

7. Describes what happens when the ACALL instruction is executed.

8. Write an instruction which is able to complement bit 7, 6, 2 and 0 of Port 2.

9. What is the advantage of using EQU directive in an assembly language program?

10. Describe what happens when an enabled interrupt is detected.

11. Describe how you would implement a 20ms delay using timer 0. Assume the clock into the timer is 1MHz.

12. What is the vector address of Timer 1?

13. What is the next available memory where the user can write a program without interfering with the interrupt vector? Give an example of program code.

14. Upon exiting Reset, what is the contents of the stack pointer?

15. Upon exiting Reset, which instruction is first executed?