EXPERIMENT NO 12

REAL TIME CLOCK

INTRODUCTION

The end of semester project for Microprocessor Interfacing and Programming Lab was given to us via a Project statement. The objective of the project was to design and implement a Digital Clock with a 2x16 LCD for displaying messages and a 5x7 Dot Matrix, along with a 4x3 Keypad for data entry.The aforementioned digital clock was to have a number of additional function besides displaying the current time. Upon pressing the ‘\*’, anoption to update the current time was to be shown. The user would then enter the time and the time would be changed. Furthermore, pressing the ‘#’ key would initiate an option to set the time for the alarm. When the timer would reach the set time, a message indication the alarm going off would be shown on the 2x16 LCD.

OBJECTIVE:

● To consolidate the knowledge of previous experiments to understand and program a

complex problem.

EQUIPMENT/TOOLS:

● ATMEL ®89C52 Microcontroller

● Microprocessor Trainer Board

● PC

PROCEDURE:

The crux of the Digital Clock was the timer0 in the 8051 microcontroller, which counts the time passed. The ISR involved setting the values of the TH0 and TL0 values which combined to forma 16 bit value. After every 60 seconds or 1 minute, the values in the register R5, which represented the rightmost digit, was incremented until it reached 9. At this point the value in the next register R4, whichrepresented the next digit from the right was incremented, and so on with the hours.The LCD and the Dot matrix displays were then used to show the messages and the current time respectively. The following piece of code shows part of the LCD interfacing:

ISSUES:

no issue found regarding this lab.

CONCLUSION:

IN this lab we learn to write a program of clock which will display time in the HyperTerminal program on the PC.

POST LAB:

Question 1 (Answer)

Using the timers, we can generated the exact value of delay which is required. But, using nested DJNZ, we can only generate an approximate value of delay.

Question 2 (Answer)

To start the timer, firstly we have to clear the TFx flag to avoid any junk value and then timer gets stopped because of overflow service routine TFx, this bit gets one when an overflow occurs and it gets clear again while exiting from ISR.

In Timer1 overflow interrupt service routine, this bit will get cleared automatically while exiting from ISR

Question 3 (Answer)

There are two types of timer, one-shot timers, and auto-reload timers. Once started, a one-shot timer will execute its callback function only once. It can be manually re-started, but will not automatically re-start itself. Conversely, once started, an auto-reload timer will automatically re-start itself after each execution of its callback function, resulting in periodic callback execution.

Question 4 (Answer)

The 8051 has two timers/counters. They can be used either as timers to generate a time delay or as counters to count events happening outside the microcontroller.