University of British Columbia
Electrical and Computer Engineering
Electrical and Biomedical Engineering Design Studio
ELEC291/ELEC292

Laboratory 2 – Interrupts / Alarm Clock

Introduction

Interrupts are very important and useful for embedded systems. An interrupt is an event that triggers the automatic execution of a predetermined piece of code. The code used to handle interrupts is often referred as the Interrupt Service Routine (ISR). The normal or default flow of code in a microcontroller system can be interrupted by different events, for example: timers overflowing, external events, critical errors (such as division by zero or illegal op-codes), power failure, communication events, etc. In this module you will be using timer interrupts in the AT89LP51RC2 microcontroller in assembly language.

References

A51 (8051 assembler) user manual included with CrossIDE. 'A51.pdf' not only describes how to use the assembler, but also includes a detailed description of all the instructions supported by the 8051 family of microcontrollers.

AT89LP51RC2 Microcontroller Datasheet. Information about timers and other hardware peripherals in the AT89LP51RC2 can be found in the datasheet.

Laboratory

1) Provided in the web page for the course is the example "ISR_example.asm".   This demonstration program carries three 'concurrent' tasks:

a) Increments (when P0.0=1) or decrements (when P0.0=0) a BCD variable every half second using an ISR for timer 2;

b) Generates a 2kHz square wave at pin P1.1 using an ISR for timer 0; and

c) in the 'main' loop it displays the variable incremented/decremented using the ISR for timer 2 on the LCD and resets it to zero if the 'BOOT' pushbutton connected to P4.5 is pressed.

Download, compile, and load the program into the AT89LP51RC2 microcontroller system and verify its correct operation using an oscilloscope or multimeter to check the 500.000Hz square wave output of pin P1.0.

2) Write/compile/run an assembly program for the AT89LP51RC2 microcontroller system with LCD for an alarm clock. The alarm clock must display hours (12 hour mode with AM/PM indication), minutes, and seconds, using the LCD. The clock's current time (hours, minutes, seconds, AM/PM), must be settable using pushbuttons. The clock must also include a settable alarm. When an alarm is trigger, a speaker should produce an alarm sound. Use the mini speaker available in the microcontroller system parts kit for this purpose. Don't forget to add extra functionality and/or features for bonus marks! Submit a short movie demonstration of your microcontroller system working as an alarm clock and your assembly code to Canvas where it will be graded by the TAs.