

THE CITY COLLEGE OF NEW YORK  
Department of Electrical Engineering

EE425 Computer Engineering Laboratory (1XB) – Summer 2016

---

**Experiment 3 – Interrupt Service Routine (ISR)**

**Objective:** This experiment is designed to show: the handling capabilities of interrupts of the PIC18F4520 microcontroller; and the ability of students to manipulate the behavior of the microcontroller. Use the microchip to detect external interrupts using the appropriate controller pins.

**Specific Tasks:** The microcontroller should be wired and programmed in a way that it executes the following tasks (READ CAREFULLY):

1. On start, the pin RD2 must output a square wave with duty cycle=50% and a half-period of 41ms. The wave must run indefinitely.
2. When a low priority (LP) interrupt event occurs:
  - The D4, D5, and D6 LEDs must start blinking in the sequence shown in the table below. The sequence must first run up from STEP 1 to STEP 8 and then back down from STEP 8 to STEP 1. There must be a 1 sec delay between steps.

STEP	D4	D5	D6
1	off	off	off
2	on	off	off
3	off	on	off
4	on	on	off
5	off	off	on
6	on	off	on
7	off	on	on
8	on	on	on

- All other activities must stop: pulse train from pin RD2 must be stopped.
  - Upon returning to the main program, the pulse train from pin RD2 must resume execution.
3. When a high priority (HP) interrupt event occurs:
    - All LED activity must stop (turn off LEDs), pulse train from pin RD2 must be stopped, and an A/D conversion process must start.
    - The HP-ISR should behave as an electronic lock: the A/D conversion must run indefinitely until the result of the conversion equals the code (8-bits binary number) given by the user.
    - The analog input voltage must come from POT1 (check the QwikFlash board schematic).
    - Program your code so that it stores the more important registers of your program before executing the ISR.
    - The result of the conversion (8-bits) must be displayed on the LCD as shown below,

T	E	S	T	I	N	G	:
0	0	0	0	0	0	0	0

- When the conversion result matches the code, the following message must be displayed on the LCD during 3 seconds before returning to the main program:

U	N	L	O	C	K	E	D

- Upon completing the HP routine, the control must return to the program that was interrupted:
  - If it was the main program, the pulse train from pin RD2 must resume execution.
  - If it was a LP-ISR, the LEDs (D4, D5, and D6) must restart the blinking as specified in numeral 2, exactly on the step it was executing at the moment the HP interrupt even occurred

The following are some steps that you need to consider when dealing with interrupts:

- Configure the registers that control the interrupts.
- Configure the required pins as inputs (external interrupts).
- Code your Interrupt Service Routines (ISR).
- Program your code so that it stores the more important registers of your program before executing the ISR.
- In the case multiple interrupts are considered, include in your code a test of the different flags so that the cause of interruption is clearly established.
- Once an interrupt has been served, clear the corresponding flag, so that the microcontroller can accept new interrupts.