LAB Experiment 4: Interrupts and A/D Conversion

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.

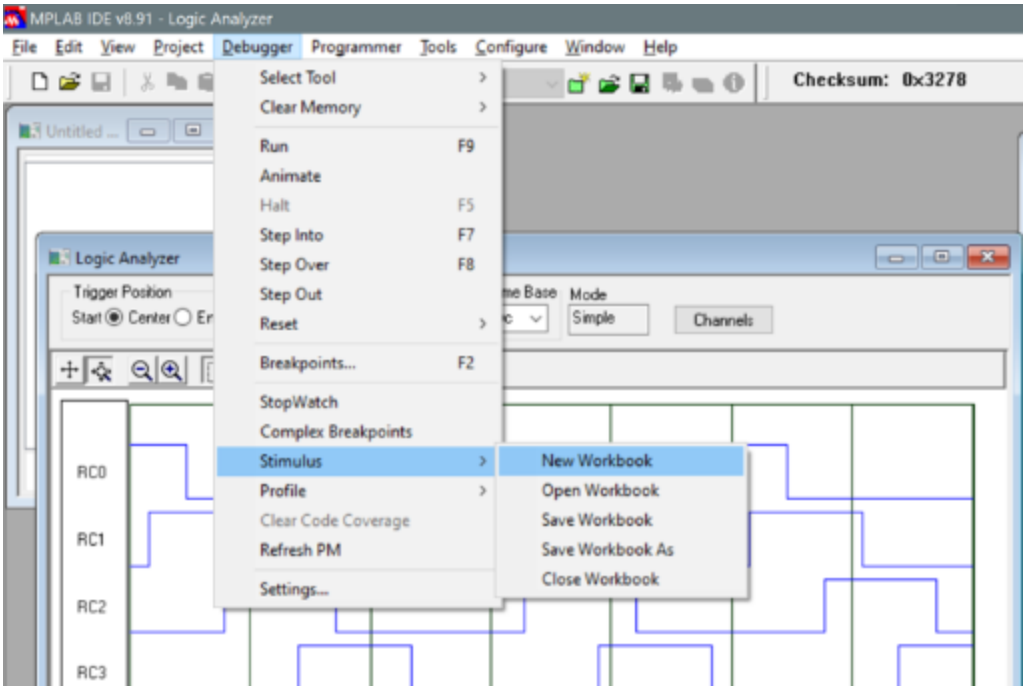
The PIC code to use

You will be provided a working copy of the code named **programForInterruptsAndADC.asm**

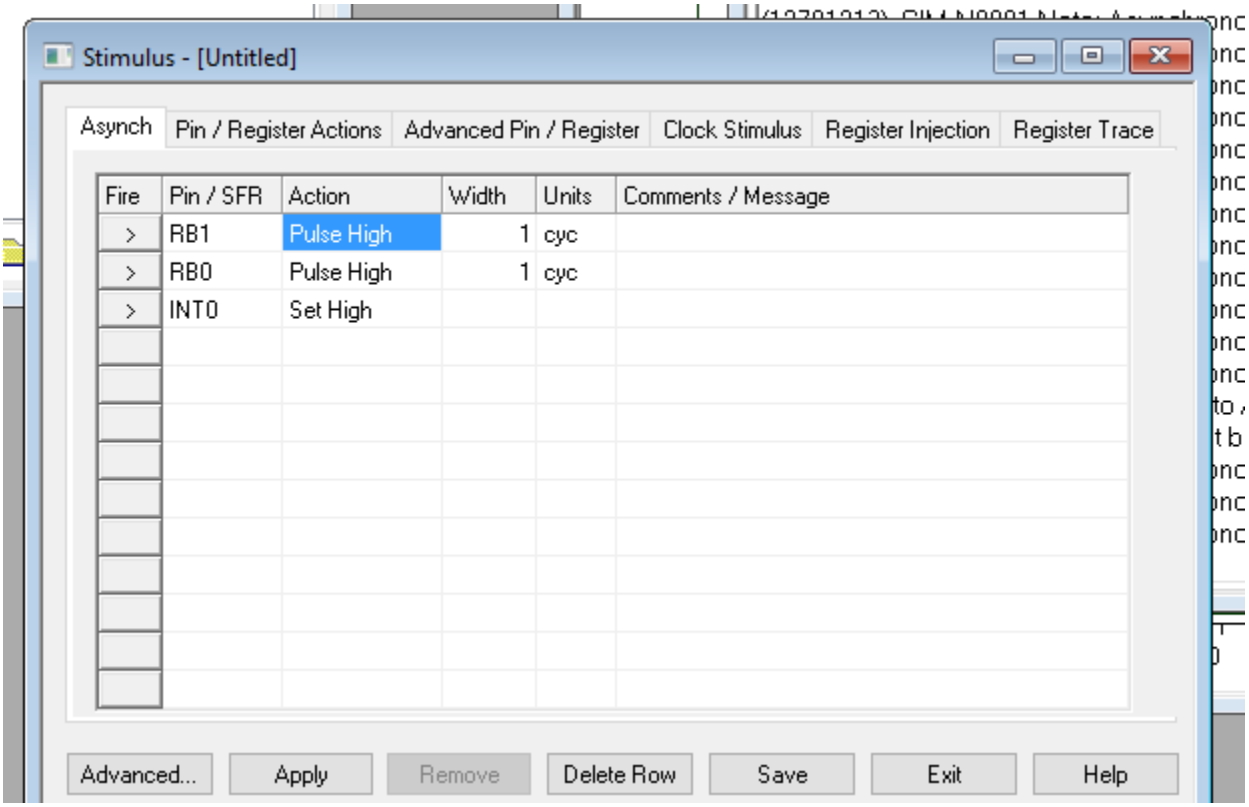
Setting MPLAB for external triggers to emulate INTERRUPTS

- After you compiled your .asm code and after setting up Debugger **(MPLAB SIM)**, select **VIEW-> LOGIC ANALYZER**

  and then select **DEBUGGER -> STIMULUS -> NEW WORKBOOK**



- Then, select Pin/SFR for setting up external PINs to trigger to execute Interrupt Service Routines (ISRs)
- For example RB1 can be used for a Low Priority Interrupt (INT1) and RB0 can be used for a High Priority Interrupt (INT0)



- During the main program execution if you want to trigger or causally speaking "to fire" an INTERRUPT event on any pin, click on the FIRE button next to it.
- Use a proper BREAKPOINT to observe the ISR execution.

Specific Tasks

The microcontroller should be wired **(in our case, our MPLAB environment must be set-up properly)** and programmed in a way that it executes the following tasks.

please read details carefully

1. On start, the pin RE2 must output a square wave with duty cycle=50% and a half-period of 0.1*ms*. The wave form 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 with a 1 milisecond delay between each step. (Port A : RA3,RA2,RA1)

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

- All other activities **must stop**: pulse train from pin RE2 must be stopped.
- Upon returning to the main program, all LEDs must be turned off and the pulse train from pin RE2 must resume execution.

3. **When a high priority (HP) interrupt event occurs:**

- All LED activity must stop (turn off LEDs), pulse train from pin RE2 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) which is eight 0s (00000000).
- The analog input voltage must come from POT1 (check the QwikFlash board schematic). [Of course , we will pretend it is the case]. ***Since we don't have a real analog input source, we will complete the High Priority Interrupt ISR by reading a ZERO at the ADRESH register.***
- Program your code so that it stores the more important registers of your program before executing the ISR.

• 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 RE2 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 event 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 you Interrupt Service Routines (ISR).
- Program your code so that is 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.
- ***Refer to Ch. 9 of the textbook and also Ch. 9 in the microcontroller datasheet for more details about interrupts.***

Your Tasks

1. Fire and experiment a LOW Priority Interrupt execution , observe the ISR execution
2. Fire and experiment a HIGH Priority Interrupt execution , observe the ISR execution
3. Fire a LOW Priority Interrupt execution , and right after that trigger (fire) a HIGH Priority Interrupt and observe what happens , explain.