Embedded and Realtime Systems

<u>Laboratory 1</u> Name:____________________________

<u>INSTRUCTIONS</u>

This laboratory should be completed within the 3 hours allocated and submitted to Moodle, as a single file, at the end of the lab. If you have multiple files archive them together e.g. zip them into one file before submitting. Be sure to include your names on all your work.

Create a document to answers to parts 1, 2, 3 and 5 of the laboratory exercise. Be sure to include a copy of your source code with your answers, either as a separate file or append at the end of your answers. Get a Demonstrator to verify part 6 of the lab.

<u>Copying will be penalised</u>

<u>DETAILS</u>

Figure 1 below depicts a basic configuration of an embedded system based around the PIC16F877A. This system is a basic alarm system, the operation of which is given in Table 1.

The circuit consists of the following:

Fosc :	4 Mhz
ARM :	Switch Input on RB0 – (KEY1 on PIC-DIP40)
ALARM:	Switch Input on RB1 – (KEY2 on PIC-DIP40)
TAMPER :	Switch Input on RB2 – (KEY3 on PIC-DIP40)
LED1 :	Green LED Output on RC0
LED2 :	Red LED Output on RC1
SPK :	Speaker Output (1khz Tone) on RC2.

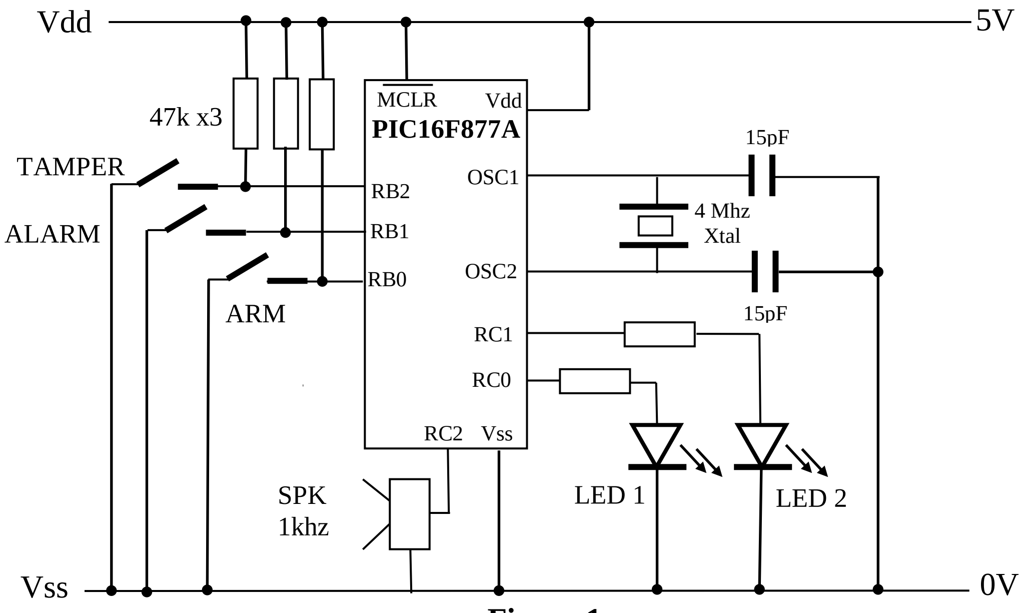


Figure 1

The desired function of the circuit is given in Table 1.

TAMPER	ARM	ALARM		LED1	LED2	SPK-1khz*
1	1	1		ON	OFF	OFF
1	1	0		ON	OFF	OFF
1	0	1		ON	OFF	OFF
1	0	0		OFF	ON	ON
0	1	1		ON	ON	ON
0	1	0		ON	ON	ON
0	0	1		ON	ON	ON
0	0	0		ON	ON	ON

Table 1

*The speaker is required to generate a tone. Look at TwoTone.c sample code.

Laboratory Exercise

1. Try simplifying the logic requirement of Table 1 so as to minimise the code requirement. **(10 marks)**

2. Using Pseudo code, outline the design of the program for the above. **(10 marks)**

3. List the Special Function Registers associated with this program task, and detail the bit configuration for each. **(10 marks)**

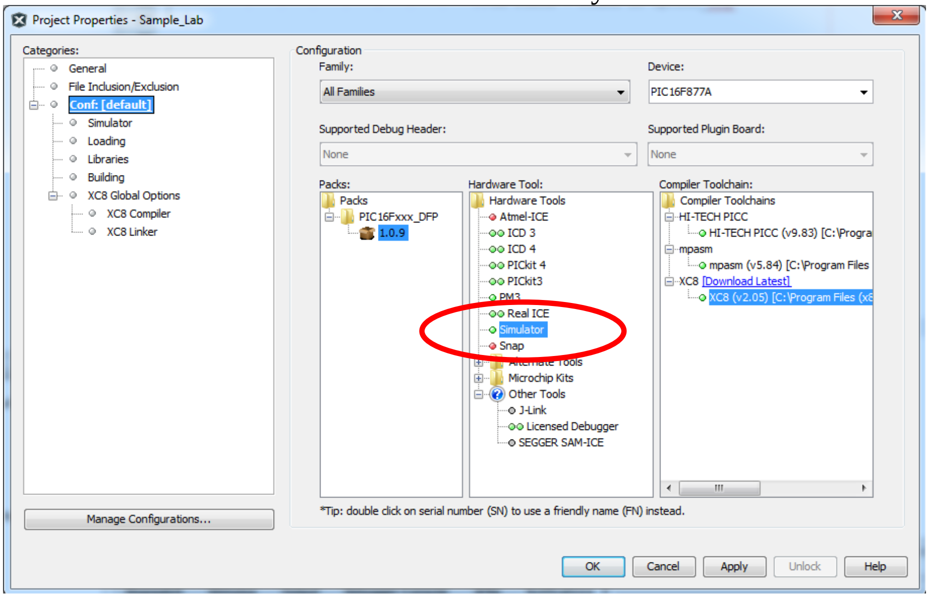
4. Using MPLAB create a project and write the C program for the above. **Include good comments and a title section for your code. (35 marks)**

5. Verify the function of the program using the MPLAB Simulator. Document your **step by step** verification procedure. **(15 marks)**

6. Using the PICkit 3, download your program to the PIC DIP-40 board and verify that your program works as expected. **Get a demonstrator to sign this off. (20 marks)**

Notes:

1. Be sure to layout your code clearly and to annotate (comment) your code fully.
2. Obvious copying will be penalised.
3. Ensure that the configuration bits are set in code.
4. Ensure your code Builds successfully. Any change to your code will require it to be Built again.
5. To verifying your program, use the MPLAB X Simulator as the debug tool. To select the Simulator, from the menu “File> Project Properties”



Project Properties - Sample_Lab

Family: All Families

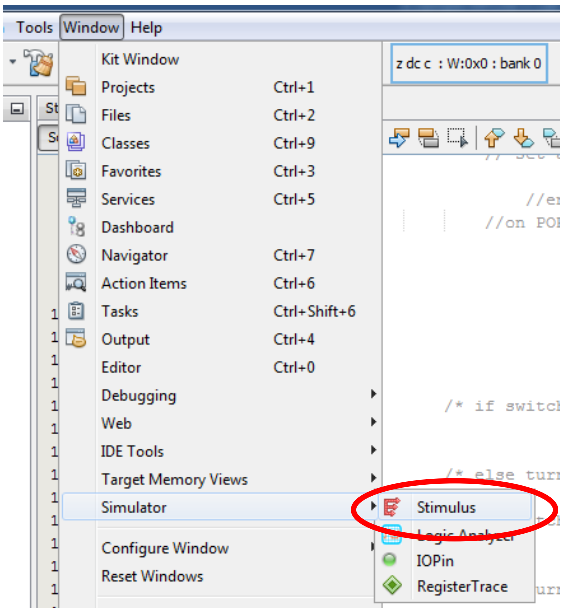
Device: PIC16F877A

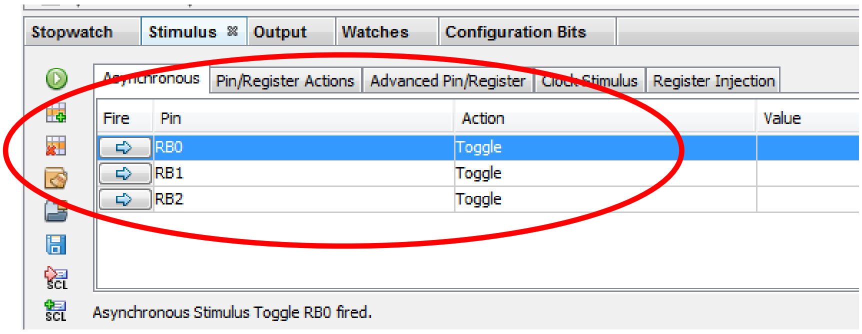
Supported Debug Header: None

Supported Plugin Board: None

*Tip: double click on serial number (SN) to use a friendly name (FN) instead.

6. To simulate switch inputs using the MPLAB X Simulator use the Asynchronous Pin Stimulus. Choose ***Stimulus.*** Add the two rows shown in the diagram. To toggle the pin click on the -> to the left of the pin name. Note that the pin won’t change until the next instruction is executed.

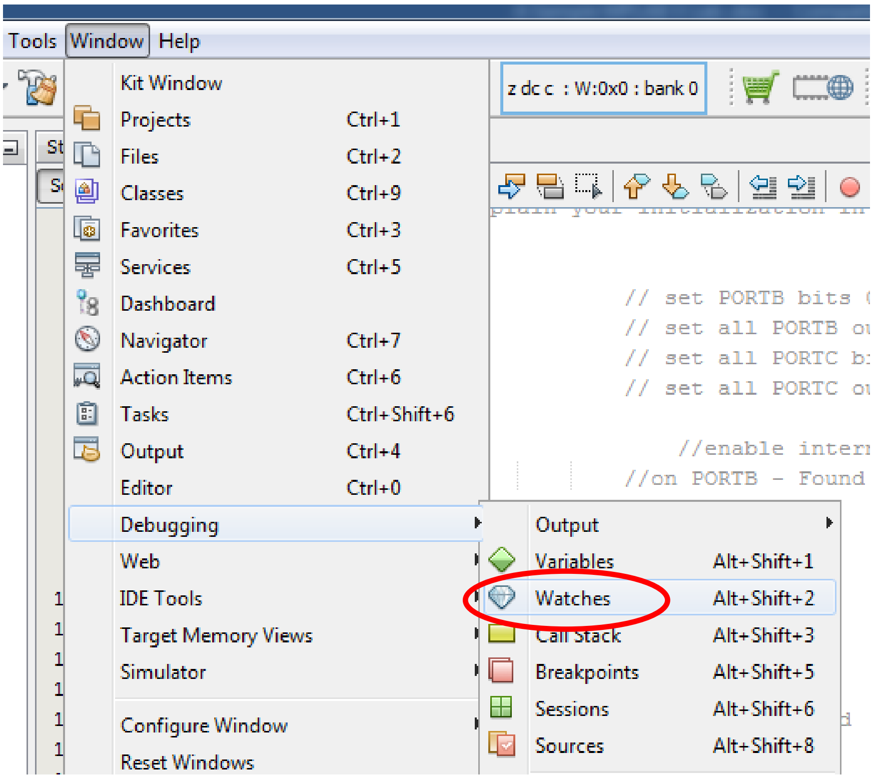




Fire	Pin	Action	Value
⇨	RB0	Toggle	
⇨	RB1	Toggle	
⇨	RB2	Toggle	

Asynchronous Stimulus Toggle RB0 fired.

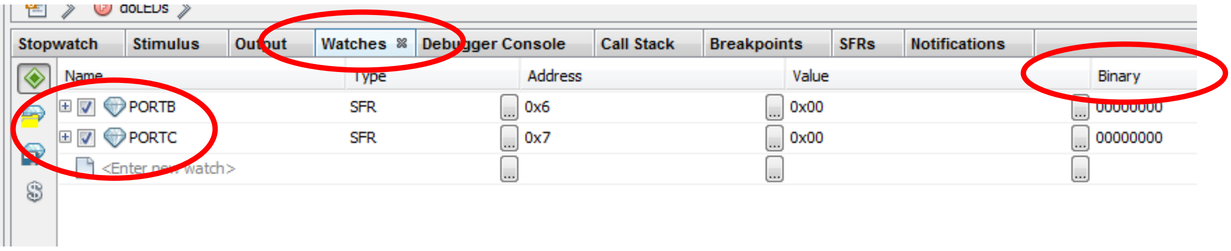
7. Use the Watches window to monitor Special Function Registers (FSRs). Select menu “Window>Debugging>Watches”



```
                  // set PORTB bits (
                  // set all PORTB ou
                  // set all PORTC b:
                  // set all PORTC ou

                      //enable interr
                  //on PORTB - Found
```

8. Then add your desired SFRs to each row. Pressing <enter> will give you the list of SFRs. You can modify the colums to include “Binary” by right clicking this row.



Name	Type	Address	Value	Binary
☑ PORTB	SFR	0x6	0x00	00000000
☑ PORTC	SFR	0x7	0x00	00000000
<Enter new watch>				

9. For final verification, connect your hardware (PIC DIP40) and select the PICKit 3 as the debug tool. Select the PICKit 3 from the menu “File> Project Properties”. Program your PIC DIP40 and run your program.