Lab #4
Interrupts

Part 2 of this lab exercise will be marked and will count towards the final coursework mark for CS1022. Submit your solution to Part 2 using Blackboard no later than 23:59 on Monday 10th April 2017. You will also be required to demonstrate your solution to Part 2 during the lab on Friday 31st March or Friday 7th April. Marks will be awarded for this demonstration.

1 Examples

Verify the operation of the **Timer**, **Button** and **Buzz** example programs from the lectures. (These can be downloaded from the Lectures section on Blackboard.)

Timer Uses interrupts raised periodically by the TIMER0 device to cause the LED, which is connected to pin P2.10, to flash. Experiment with modifying the value loaded into the match register MR0 of the timer.

Button Uses interrupts raised when the button connected to pin P2.10 is pressed to count the number of times the button is pressed. The count is stored in memory and can be examined using µVision.

Buzz Combines both button and timer interrupts. While the button is held down, a sound is emitted from the on-board speaker. When the button is released the sound stops. Timer interrupts are used to control the frequency of the sound. Experiment with modifying the value loaded in to the match register MR0 of the timer to control the frequency of the generated sound.

Each of these examples uses devices (LED, button) connected to pin P2.10. However, the examples configure pin P2.10 differently. In the **Timer** example, P2.10 is configured as an output GPIO pin and is used to control the LED. In the **Button** and **Buzz** examples, P2.10 is configured as an external interrupt pin (internally connected to the EINT0 channel on the Vectored Interrupt Controller).

The **Buzz** example uses the LPC2468's Digital to Analog Converter (DAC) to control the speaker. The speaker is hard-wired to the analog out (AOUT) pin of the DAC. The DAC outputs an analog voltage level that can be controlled programatically by writing a 10-bit (digital) value to bits 6:15 of the DAC Register (DACR). By alternating this value between 0 and a with frequency f, we can generate a sound with frequency f and amplitude a. (See Chapter 29 of the LPC2468 User Manual for further information on the Digital to Analog Converter.)

2 Rugby Match Timer

Consider a timer device used to measure the elapsed time in a rugby match. A single push-button is used to control the timer. The first time the button is pressed, the timer should begin counting the elapsed match time in seconds. If the button is pressed again the timer should pause counting. It should resume counting the next time the button is pressed, and so on. When the match time reaches 40 minutes, a buzzer should sound for five seconds.

Develop an ARM assembly language program to implement the Rugby Match Timer described above. (You may want to replace 40 minutes with a shorter match time to make testing less tedious!)

Although you can approach the problem using a single TIMER device, you may find it more convenient to use two TIMER devices (e.g. TIMER0 and TIMER1). One TIMER device might be used to increment the match time and control the duration of the final hooter. The second TIMER might be used to control the frequency of the hooter.

(Note that if you decide to use TIMER2 or TIMER3, you will need to turn them on first, as they are off by default. The LPC2468 User Manual tells you how to do this.)