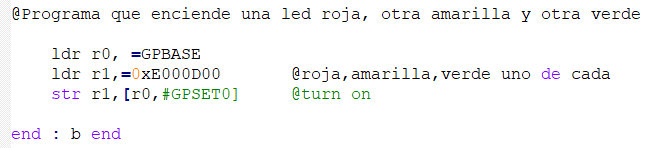
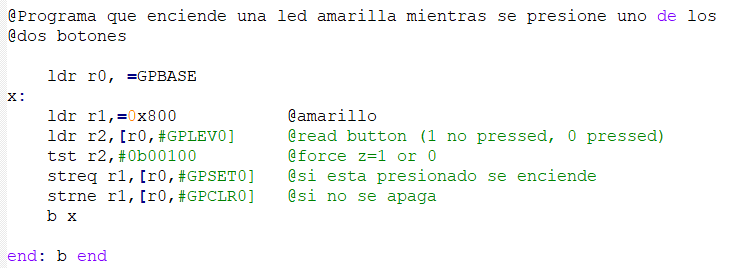
COMPU RASPBERRY

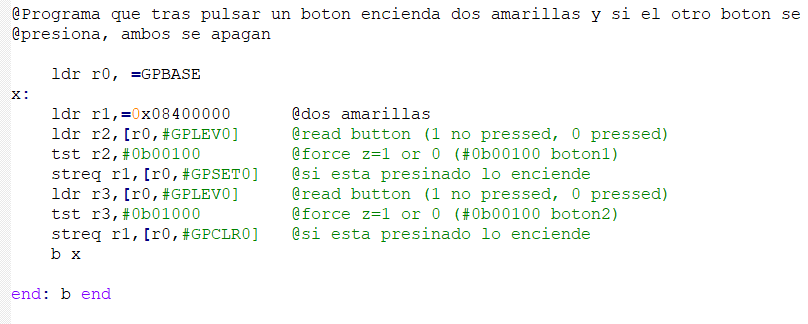
**1. Write a program that turns one red led, one yellow led and one green led of the expansion board. At startup, we know that the LEDs are all off. Denote Exer1.s this program. Remember that due to the lack of an operating system, your program has to finish by an infinite loop.**



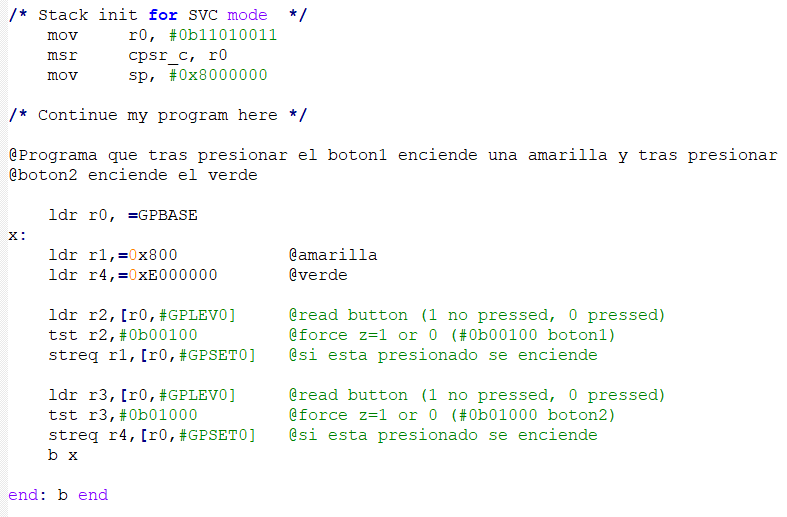
**2. Write a program that turns on one of the yellow leds while you are pressing one of the buttons, and it turns it off if you release the button.**

****

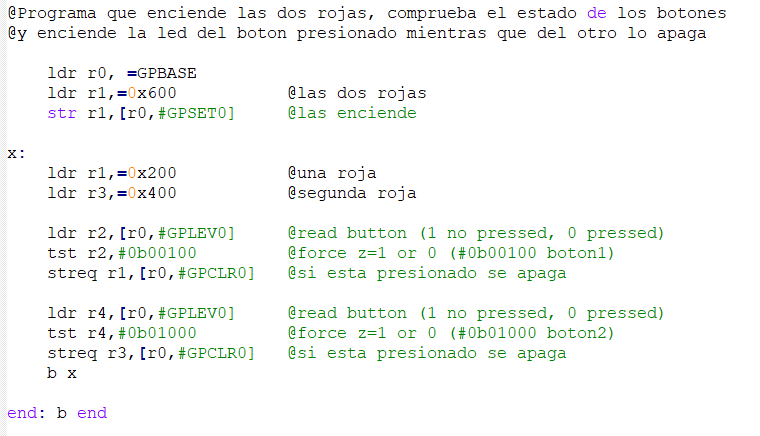
**3. Write a program that turn on the two green leds after pressing one button (edge triggered). If the other button is pressed, both leds will turn off.**

****

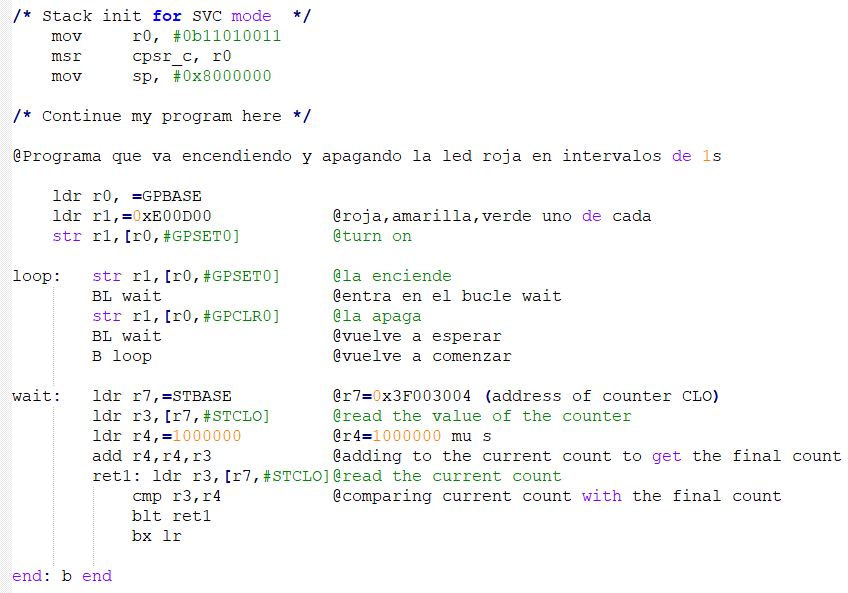
**4. Write a program that after pressing the button 1, one of the yellow leds turns on permanently, and if you press the button 2 a green led will turn on permanently.**

****

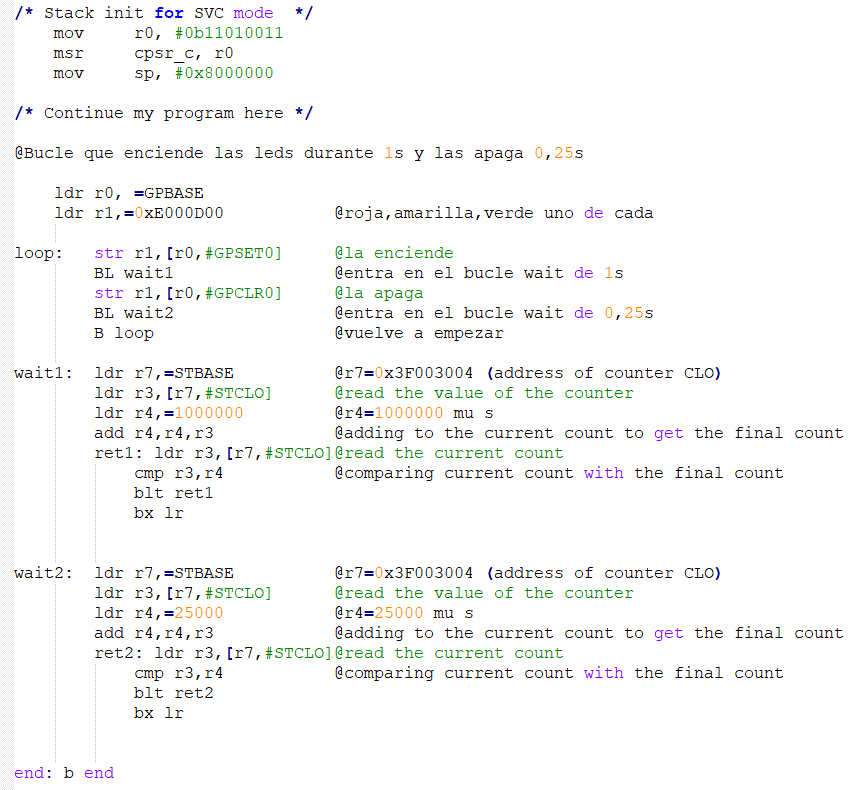
**5. Write a program that turns on the two red LEDs. After this, the program polls the state of the push buttons. Once a push button is pressed, the corresponding LED will keep on, whereas the other LED will turn off.**

****

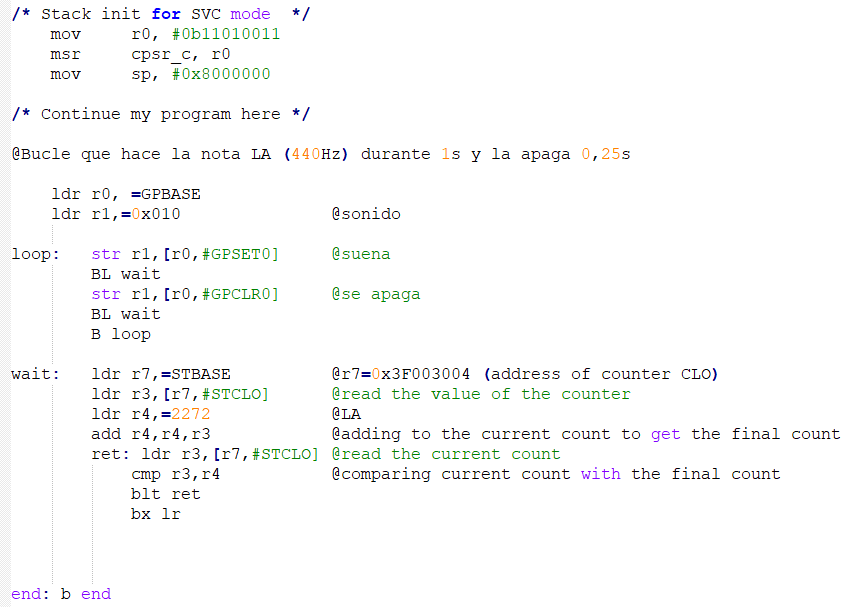
**6.Modify the program Ejer1.s to create a new one which flashes the LEDs at the rate of 1 s. ON-OFF. Insert the corresponding delay using the timer.**

****

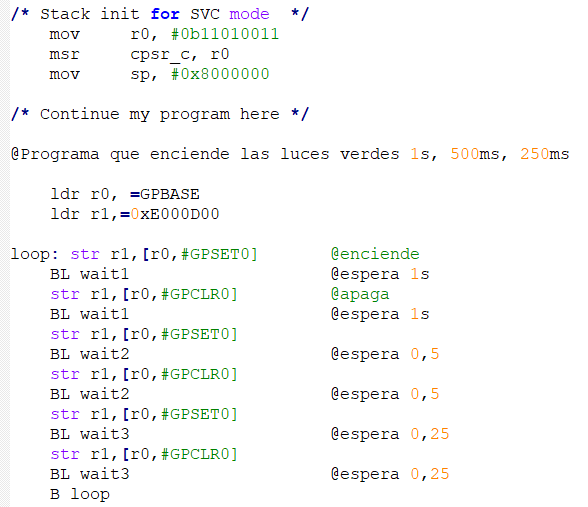
**7. Flash the LEDs at a rate of 1 s. ON, 0.25 s. OFF by using the timer.**

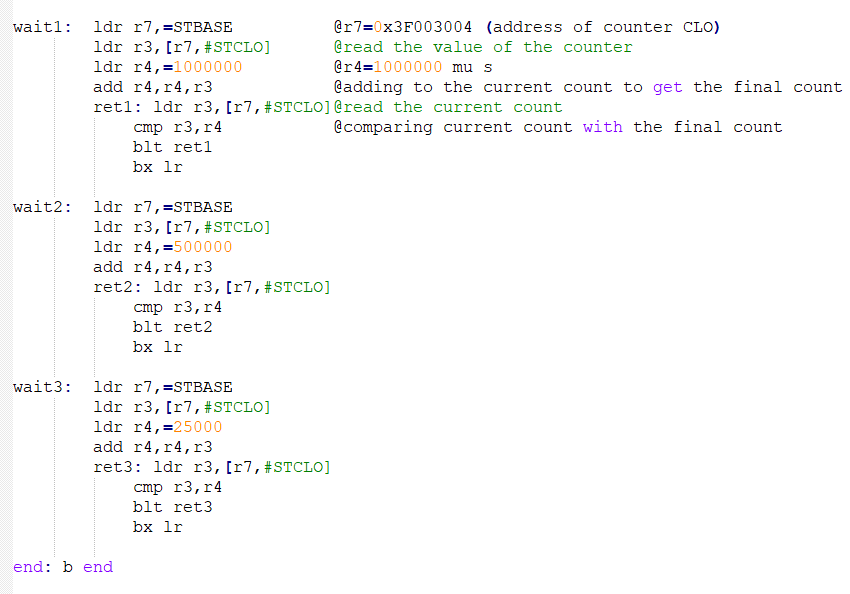
****

**8. Modify the program of the previous exercise such that instead of acts on the LED, it generates a 440Hz tone (note LA) on the speaker.**

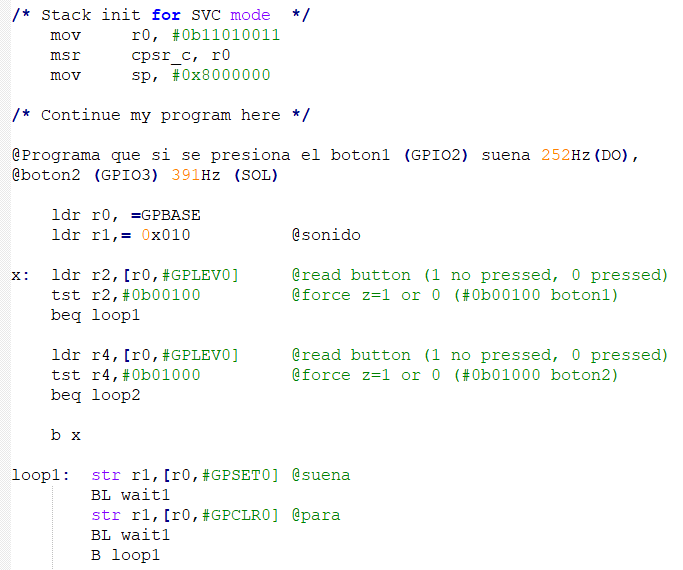
****

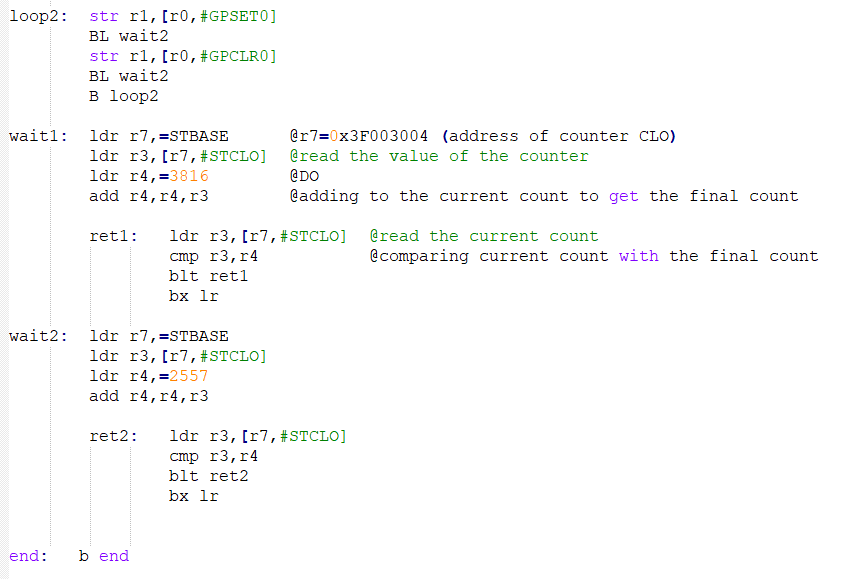
**9. Write a program that the green LEDs flashes at a rate of 1 second ON-OFF, then at a rate of 500 ms. ON-OFF and finally 250 ms. ON-OFF in an infinite loop.**

****

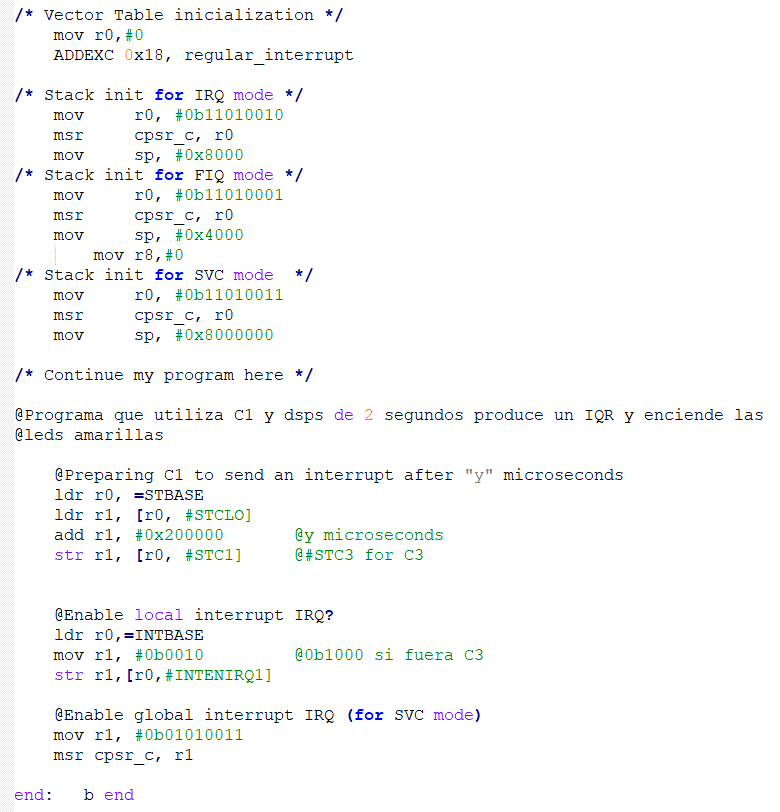
****

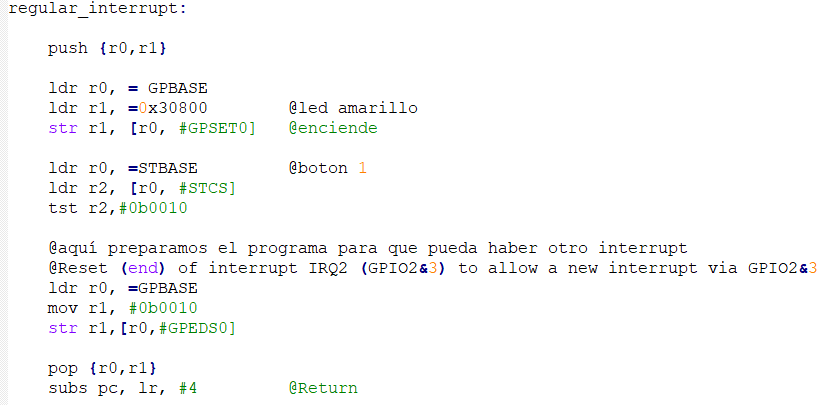
**10. Write a new program which polls both the buttons 1 (GPIO 2) and 2 (GPIO 3). If the first pressed button is the button 1, a tone of 262Hz (note DO) has to be generated. Otherwise, if the first pressed button is the botton 2, a tone of 391Hz (note SOL) is generated.**

****

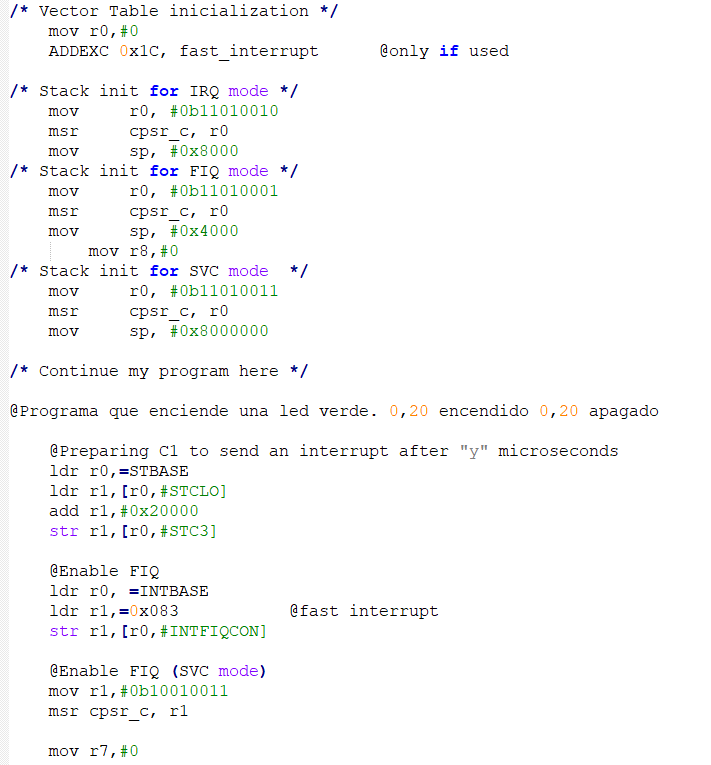
****

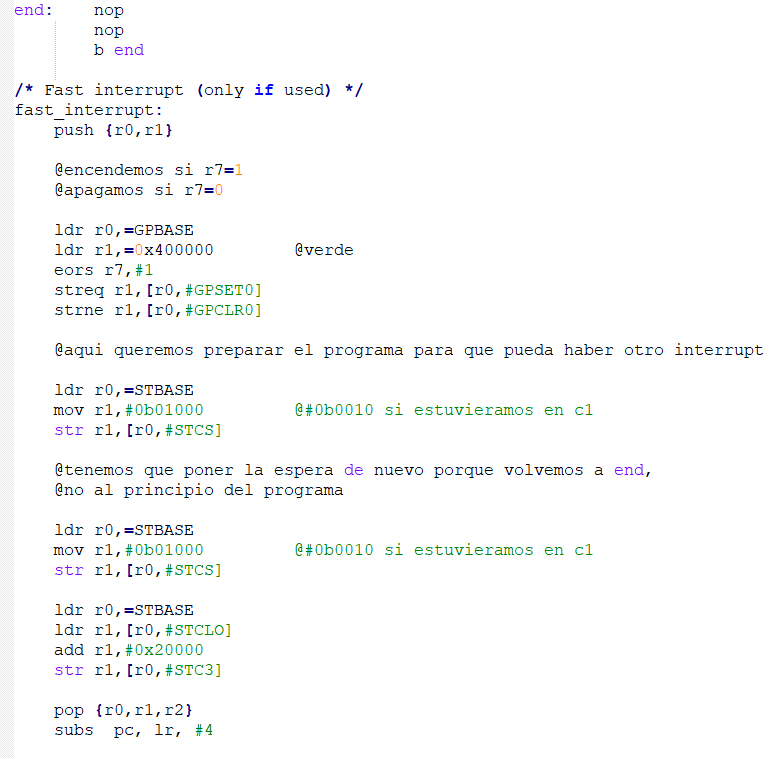
**11. Write a code (Exer11.s) which configure the timer C1 comparator such that after 2 seconds a IRQ is produced, and the corresponding handler routine turns on the YELLOW LEDs**

****

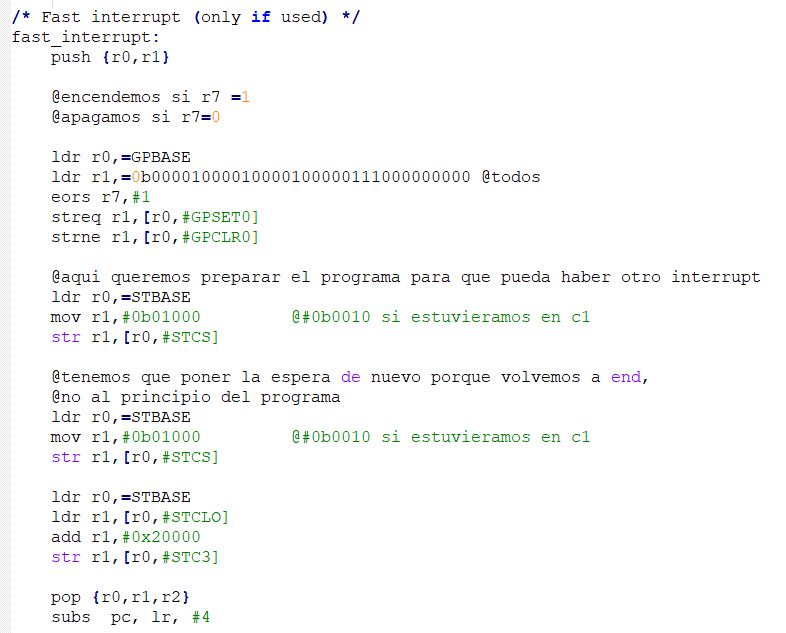
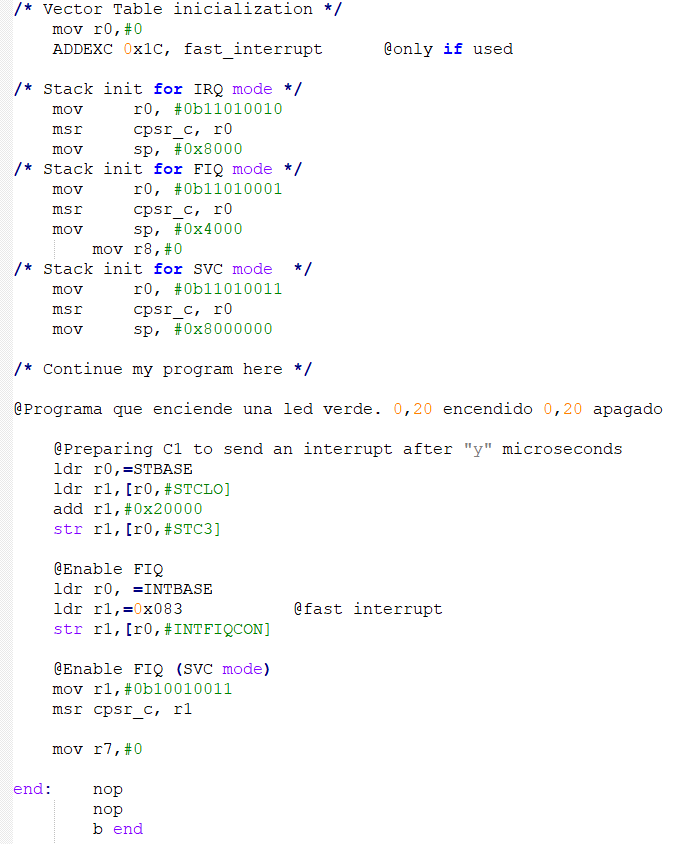
****

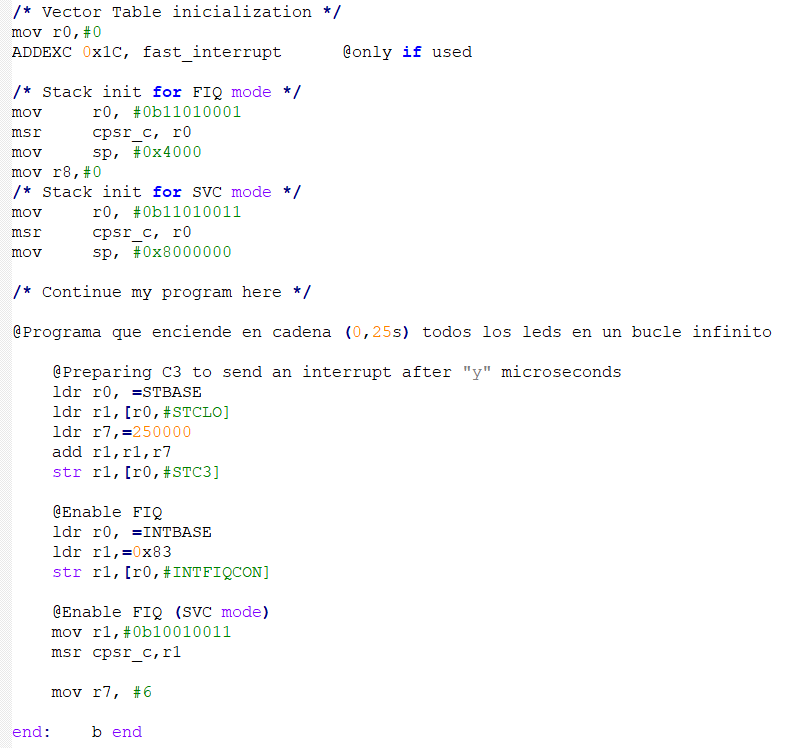
**12. Write a code (Exer12.s) to flash one of the green LEDs by interrupt. The handler routine has to re-program the comparator to provoke a new interrupt. The led has to be turned ON-OFF depending on the previous ON-OFF state. The cadence is 0.20 seconds (0.20 seconds ON, 0.20 seconds OFF).**

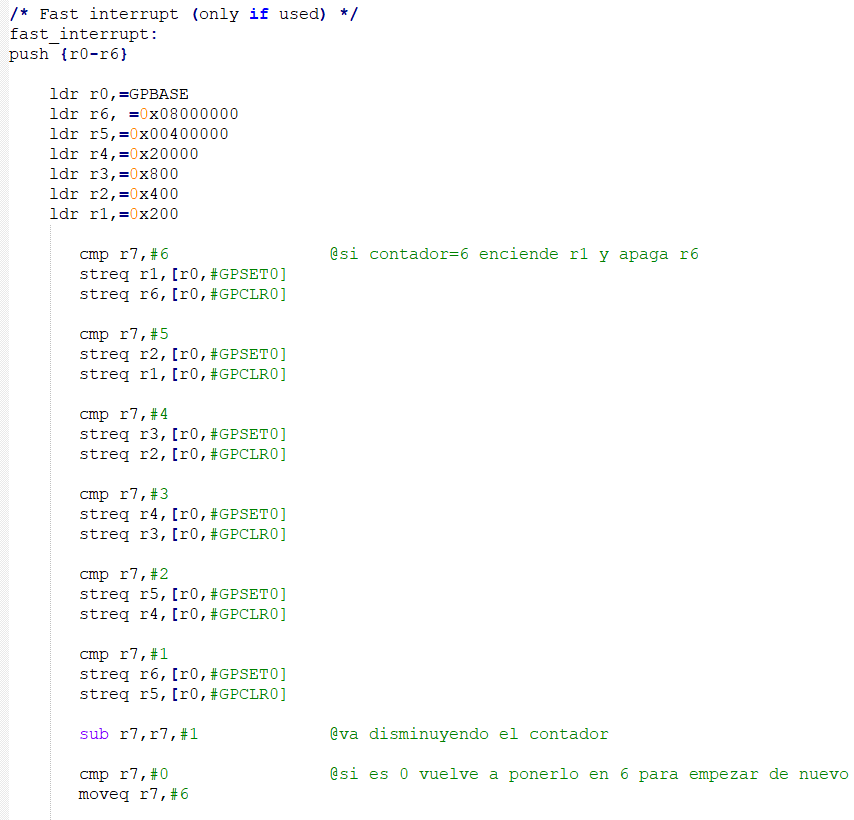
****

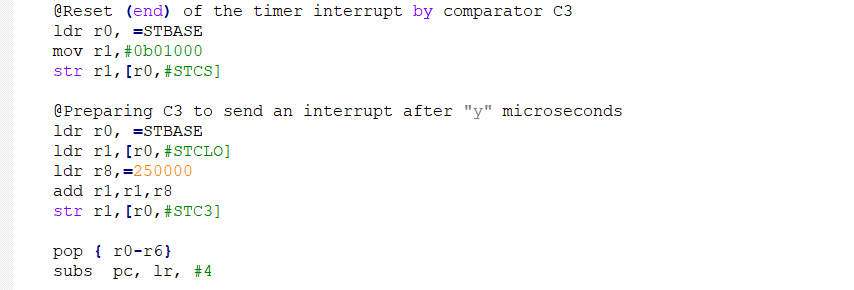
****

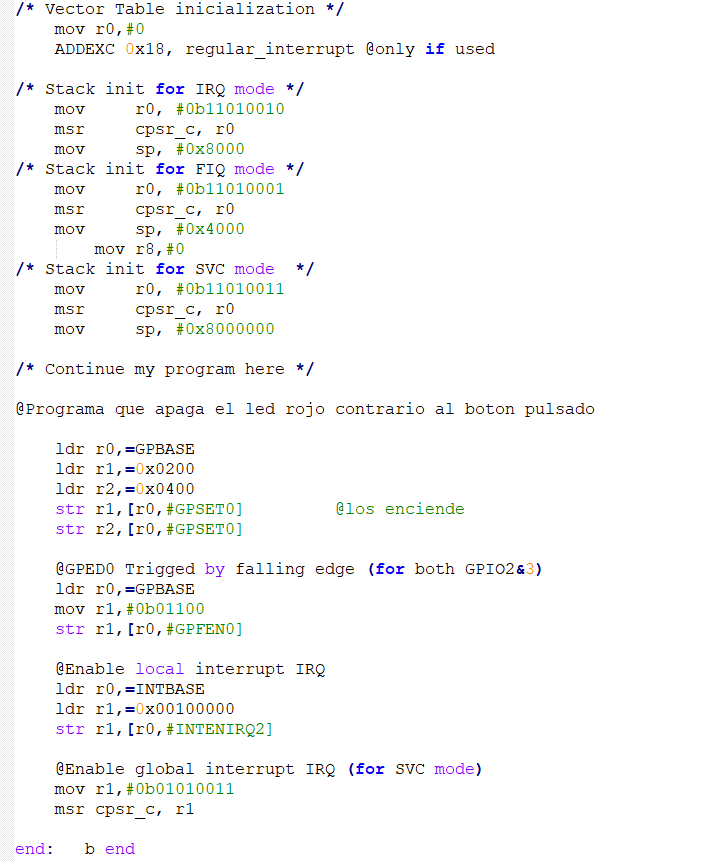
**13.The same as previous exercise, but for the 6 LEDS at the same time**

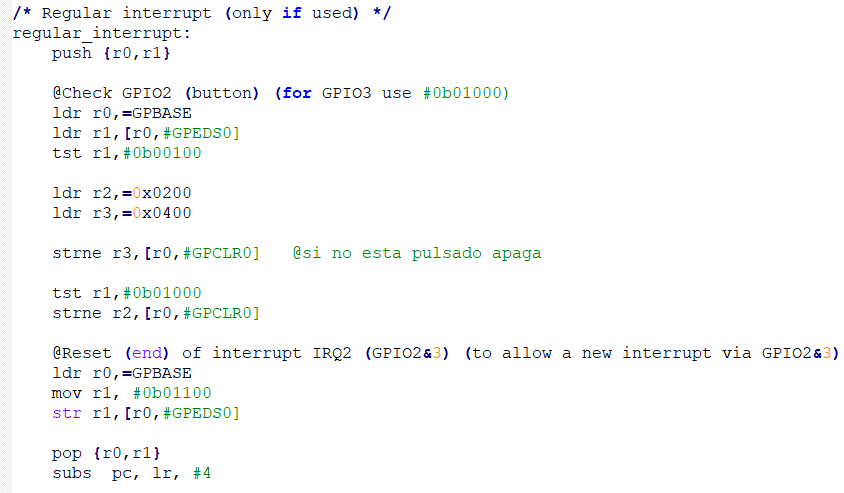
****

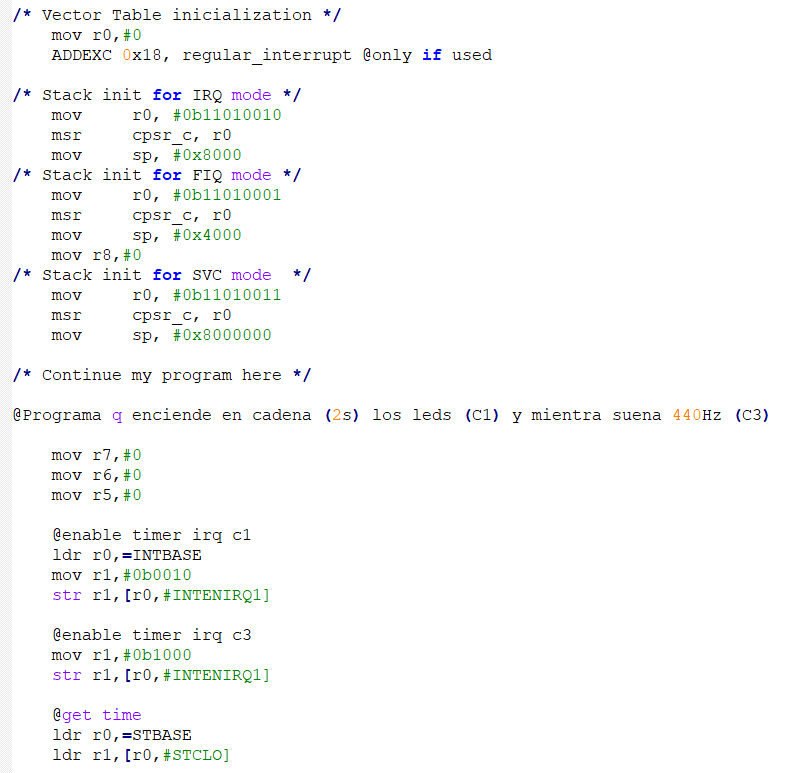
**14. Write a code (Exer14.s) that flashes the LEDs in turns with a cadence of 0.25 seconds (each led will be ON for 0.25 s.). After reaching the last LED, it will start again from the first LED forever (infinite loop). **

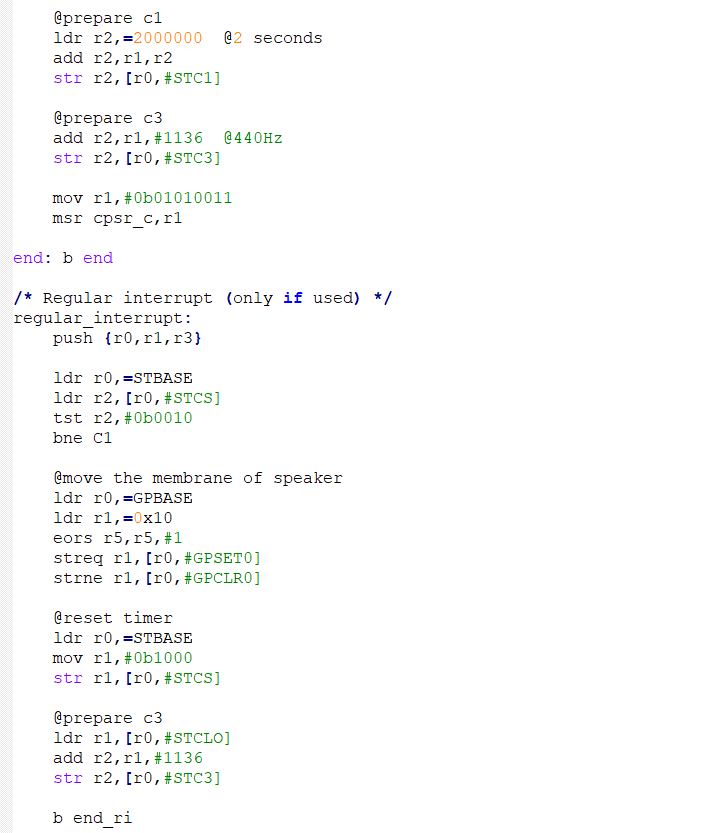
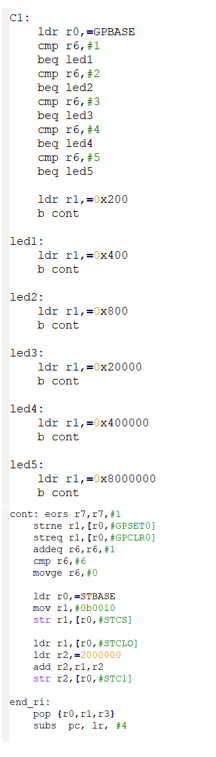
****

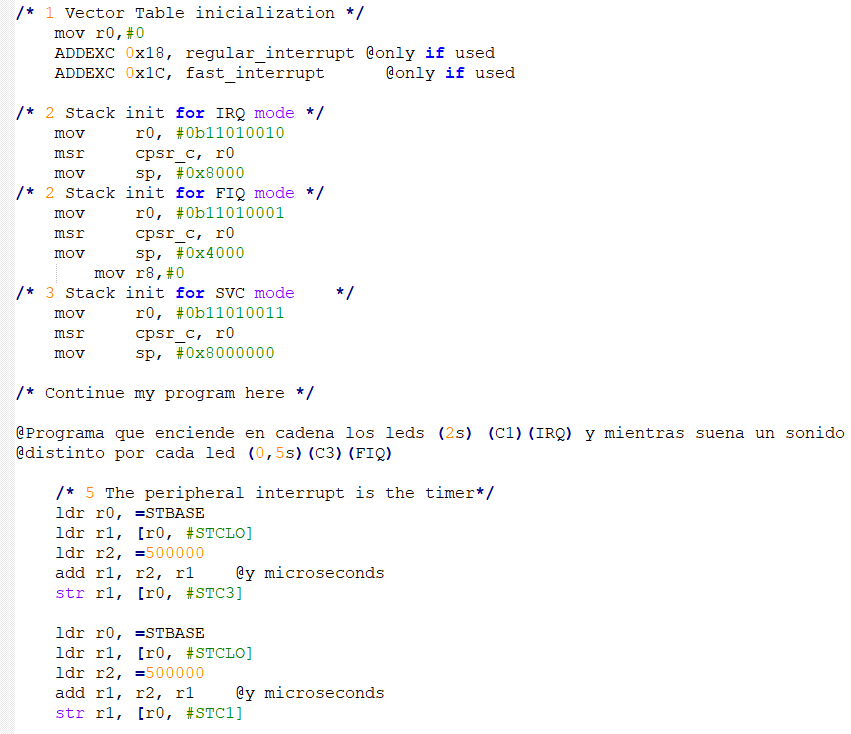
****

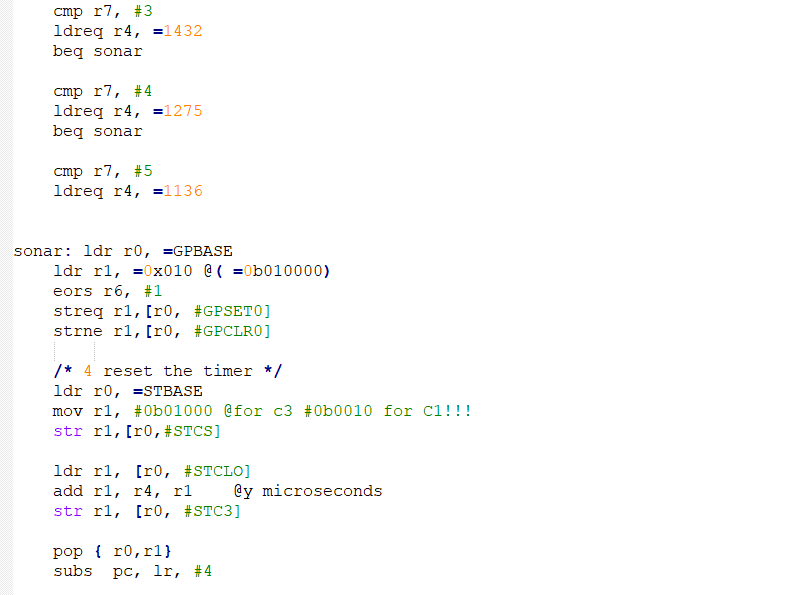
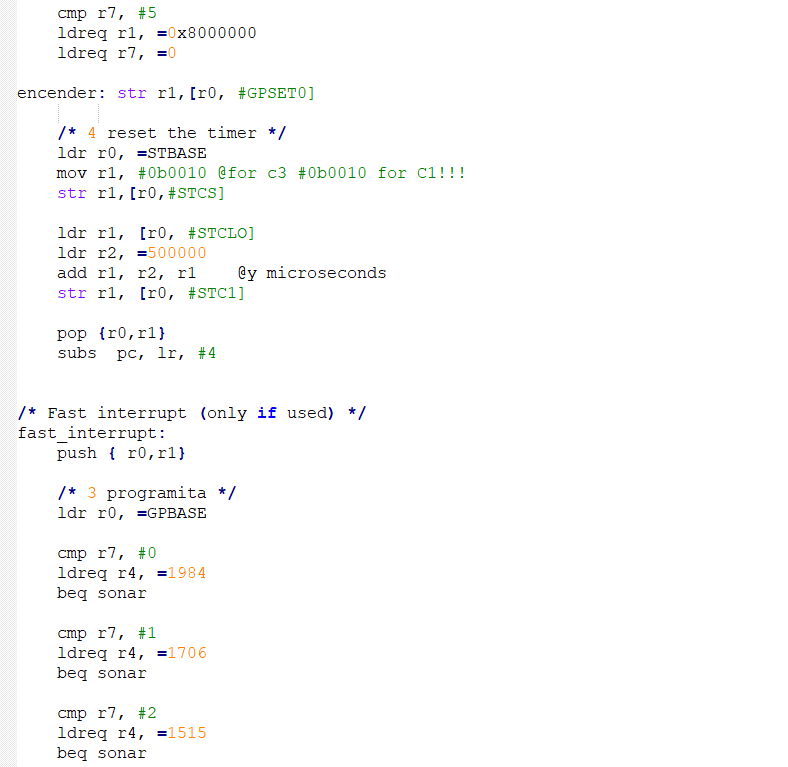
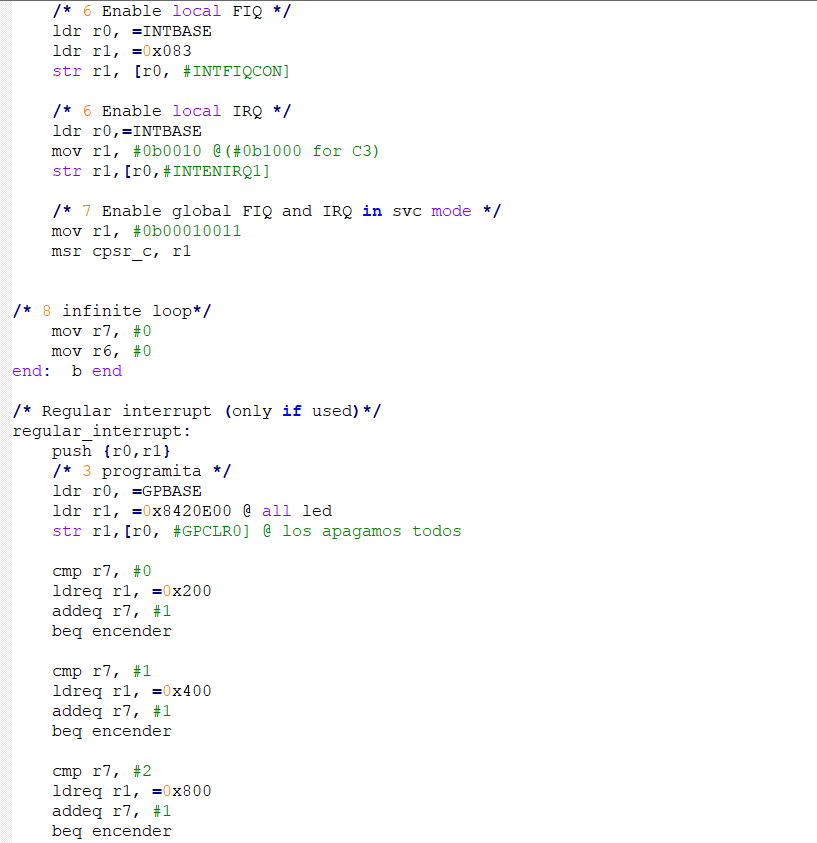
**15. Write a code (Exer15.s) that turns on the two red LEDs. After pressing any button, an IQR is generated. The handler routine has to determine what is the pressed button and keep ON only the LED of the same side (this exercise is similar to exercise 2 but with interrupts) **

****

**16. In this exercise we work with the comparators C1 and C3 and the IRQ simultaneously. With C1 we control the ON state of the LEDs with a cadence similar to the exercise 14, but with a time of 2s instead of 0.25 s. With C3 we control the speaker to produce a continuous sound of 440Hz**

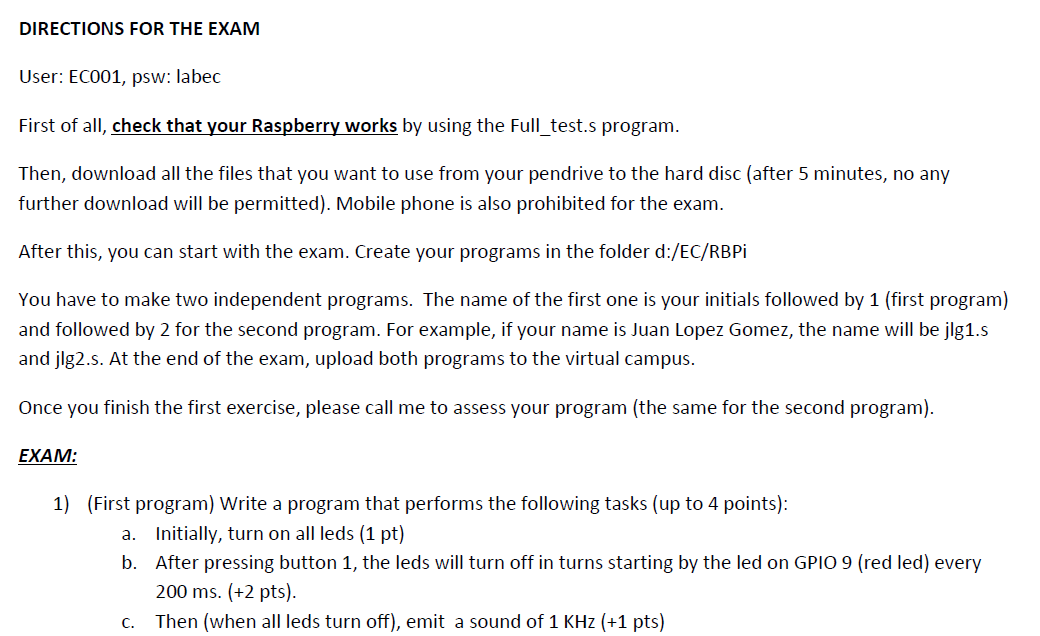
****

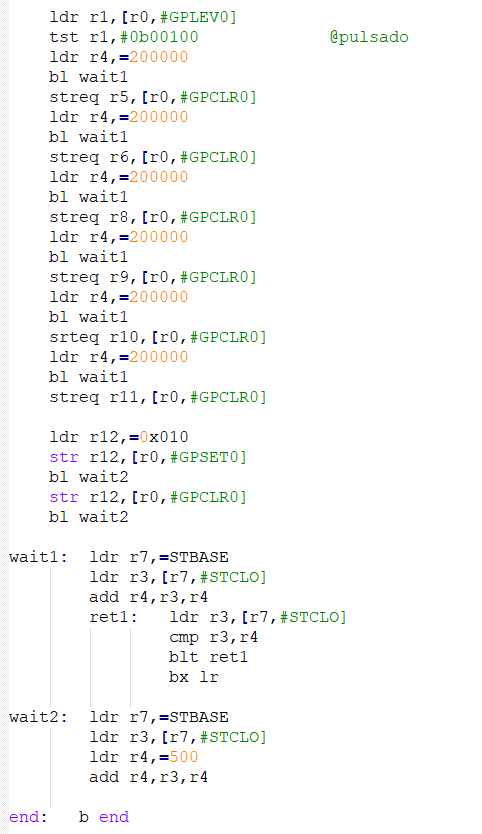
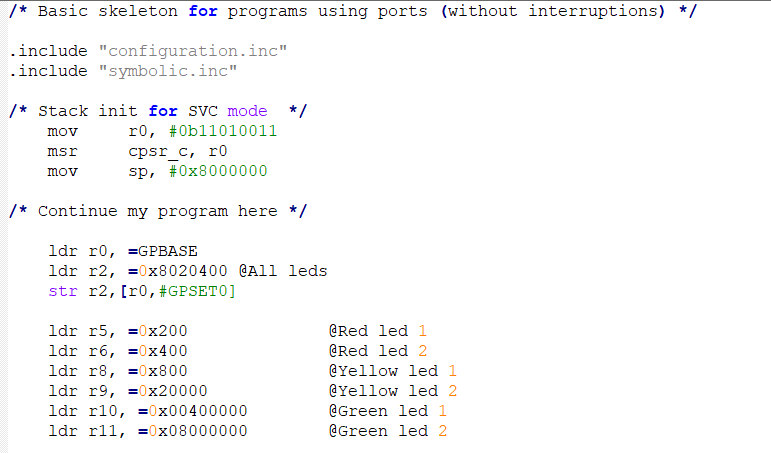
**17. Similar to the previous exercise, but every LED will be associated to a different sound. In this case, each LED (and its associate tone) will be ON for 0.5 s.. To do that, apart from the regular IRQ, we are going to use a fast interrupt (FIQ) in such a way that we have two independent handler routines. C1, which controls the sequence of lighting of the LEDs, will interrupt with a IRQ, whereas C3, which controls the speaker, will work with a FIQ. The interrupt associated to C3 has the highest priority since it will take place more frequently. Next table shows the frequency of different notes: **

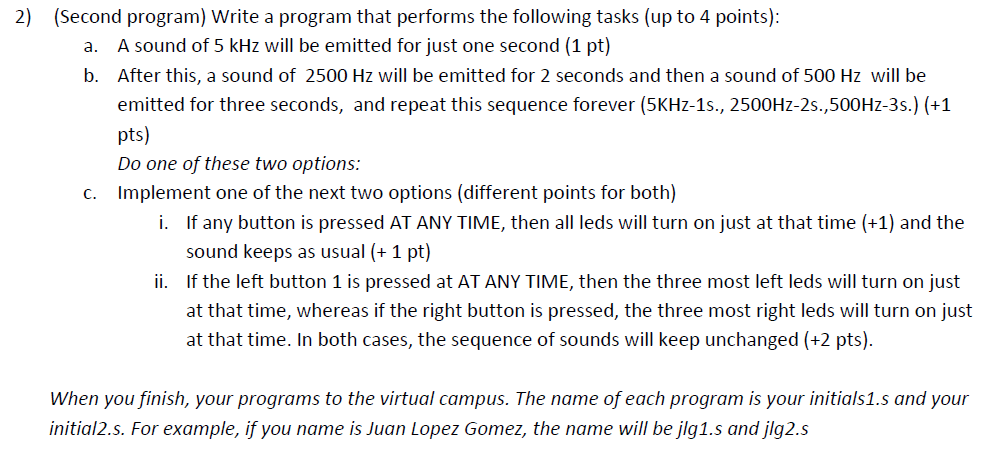
****

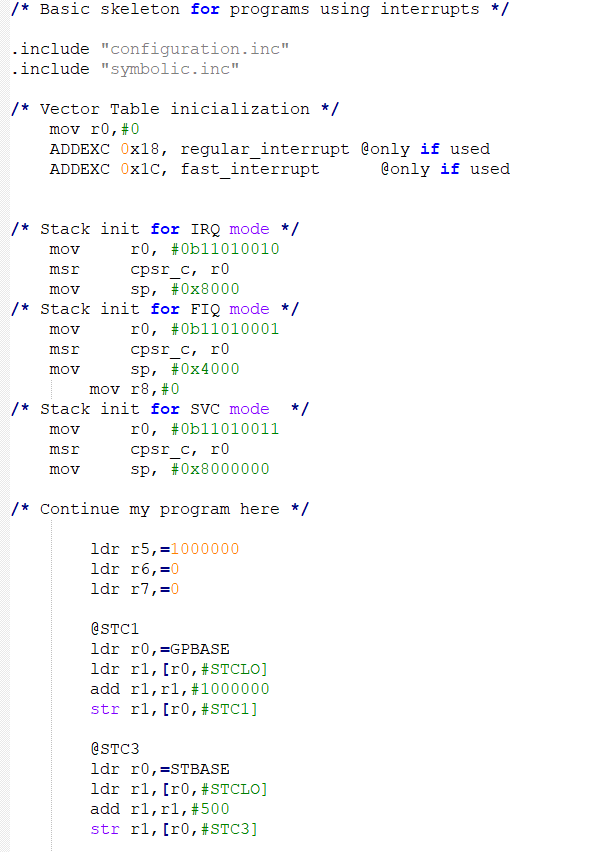
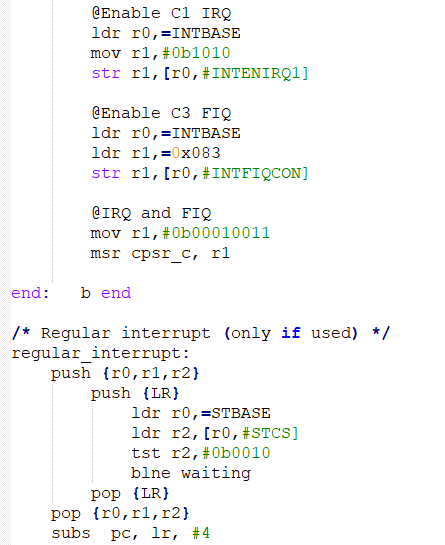
EJERCICIOS PARCIALES

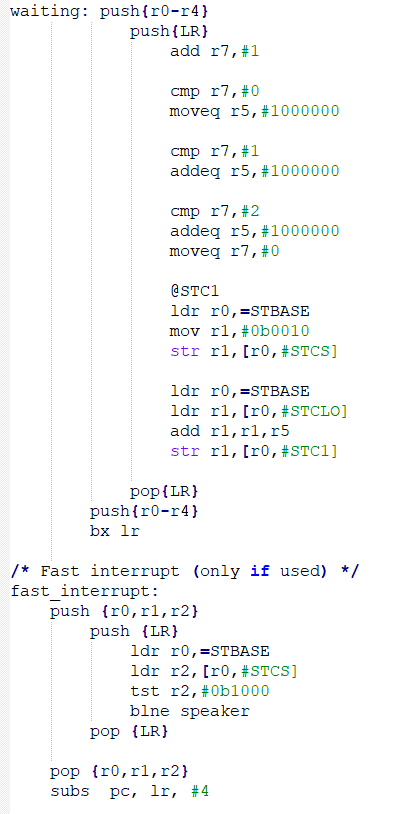
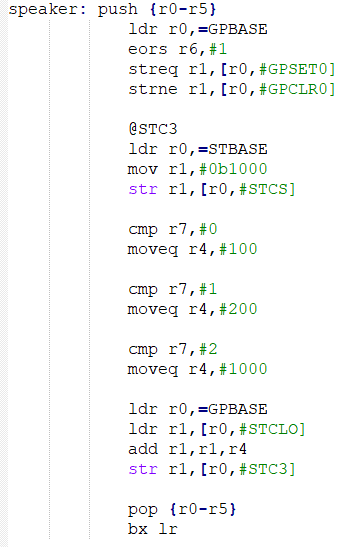
PARCIAL 2018









PARCIAL 2017

