



CUESTIÓN (2 puntos):

Escriba las instrucciones en C necesarias para realizar las siguientes operaciones, sin alterar el comportamiento del resto del sistema:

- Configurar el pin PB12 como entrada digital
- Desactivar en el NVIC las interrupciones por EXTI4 (canal 10)

PROBLEMA 1 (5 puntos):

Basándose en un STM32L152RB con $pclk$ de 32MHz, se ha de diseñar un reloj con control de la frecuencia cardíaca, que cumpla con los siguientes requisitos:

- El reloj debe mostrar (salvo alarmas) siempre la hora y los minutos en el LCD.
- El microcontrolador tiene conectado a una entrada un sensor de pulso, que cada vez que detecta un latido, emite un pulso positivo (pasa de 0 a 1 durante un tiempo y luego baja a 0), de una duración inferior al milisegundo.
- Si el pulso detectado es mayor de 140 latidos por minuto, se debe generar una alarma en forma de una señal periódica de 1KHz que atacará a un altavoz. En cuanto el pulso baja de 140, la alarma debe parar.
 - Esa alarma debe mostrar en el LCD el mensaje "PARE"
- Si el pulso detectado es inferior a 40 latidos por minuto, se debe generar una alarma, esta vez de 500Hz, que atacará al mismo altavoz. En cuanto el pulso sube de 40 latidos, la alarma debe parar.
 - Esa alarma debe mostrar en el LCD el mensaje "MEDICO"

Con estas especificaciones (de forma razonada e intentando reducir el número de recursos utilizados) proporcione:

- Diagrama de Bloques del sistema. (20%)
- Indique si utilizará interrupciones, y en caso afirmativo, cuáles y para qué. (20%)
- Indique qué periféricos utiliza y cómo los configuraría. (30%)
- Diagrama de flujo. (30%)

PROBLEMA 2 (3 puntos):

El código siguiente se ha de ejecutar en la placa STM-Discovery, y su intención es utilizar la conversión analógico/digital de un único canal mediante interrupciones, sacando el valor convertido en el LCD. Dicho código está lleno de errores, por lo que se le solicite que los localice y proponga cómo arreglarlos.



APLICACIÓN

```
1 #include "stm32l1xx.h"
2 #include "Biblioteca_SDM.h"
3
4 void ADC1_IRQHandler (void) {
5     valor = ADC1->DR;
6     ADC1->SR = 0;
7 }
8
9 int main(void){
10     unsigned short valor;
11     unsigned char texto[6];
12     Init_SDM();
13     Init_LCD();
14     valor = 0;
15
16     GPIOA->MODER |= 3<<(2*3);
17     ADC1->CR1 = 0x00000022;
18     NVIC->ISER[0] = (1 << 18);
19     ADC1->CR2 &= ~(0x00000001);
20     ADC1->CR2 = 0x00000412;
21     ADC1->SMPR1 = 0;
22     ADC1->SMPR2 = 0;
23     ADC1->SMPR3 = 0;
24     ADC1->SQR1 = 1 << 20;
25     ADC1->SQR5 = 0x00000000;
26     ADC1->CR2 |= 0x00000001;
27     while ((ADC1->SR&0x0040)==0);
28     ADC1->CR2 |= 0x40000000;
29     ADC1->SR = 0;
30
31     while (1) {
32         LCD_Texto(valor);
33     }
34 }
35
```