**NVIC.h**

Este archivo define y contiene variables para poder acceder de una forma ordenada a cada uno de los registros del NVIC.

Este registro contiene 7 funciones diferentes:

**-->NVIC\_ISER:**  Este registro solo permite habilitar una interrupción o excepción. Si se tiene una interrupción pendiente y está habilitada, esta se ejecuta según su prioridad. Por otro lado, si se da una señal que active una interrupción, el registro cambiará a un estado pendiente pero nunca se ejecutará debido a que no está habilitado este registro para esa interrupción.

**-->NVIC\_ICER:** Este registro permite deshabilitar una interrupción o excepción.

**-->NVIC\_ISPR:** Podemos cambiar por software el estado de una interrupción o excepción a pendiente, para que esta sea ejecutada por el NVIC.

**-->NVIC\_ICPR:** Permite limpiar la bandera de pendiente de una interrupción o excepción.

**-->NVIC\_IABR:** Este registro solo permite leer el registro del estado pendiente de una interrupción. Si es 1 la interrupción está activa.

**-->NVIC\_IPR:** Este registro permite poder asignar las prioridades a los vectores de interrupción

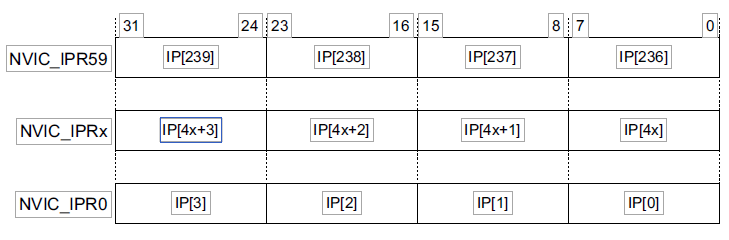


Figura 1. Mapa de campos para Prioridad en el NVIC.

Cada NVIC\_IPRx consta de 4 campos que contienen un vector de interrupción cada uno. A cada uno de estos vectores se le puede asignar una determinada prioridad, siendo 1 la más alta y 15 la más baja. Para poder asignar la prioridad contamos de un registro de 8 bits para cada campo, sin embargo, los primeros 4 bits son Reservados, es decir, que la prioridad se tiene que asignar en los 4 bits más significativos.   
Como ejemplo trabajaremos con el NVIC\_IPR0. Si a este le asignamos el valor 0x40302010. El vector de interrupción IP3 tendría la prioridad 4, el IP2 la prioridad 3, el IP2 la prioridad 2 y el IP0 la prioridad 1.

**–NVIC\_STIR:**  Software Triggered Interrupt Register, este registro permite poder generar una interrupción por software. Para eso necesitamos escribir el vector de interrupción que queramos activar, por ejemplo, si escribimos 0x03, se activará la interrupción IP3 respetando su prioridad.

Este registro necesita de privilegios para poder ser utilizado, para esto, necesitamos poner en 1 el bit USERSETMPEND en el registro SCR (System Control Register). Sin ello, no podremos acceder ni configurar este registro.

**NVIC.c**

En esta parte se encuentra contenida una única función que permite poder habilitar una interrupción, borrar la bandera de estado pendiente y además asignarle una respectiva prioridad.   
  
La función es **void NVIC\_SetCFGR(uint8\_t position, uint8\_t priority)**

**Position:** Esta variable se llena con el vector de posición para ese periférico. El valor va del 0 al 239, ya que solo tenemos 240 vectores de interrupción.

**Priority:** Indica la prioridad para esa determinada interrupción. En este caso va un número de 0 a 15.

Siendo el 0 la más alta prioridad y el 15 la más baja.

Esta función permite poder escribir a través de apuntadores en los registros que permiten la habilitación correcta de una interrupción, esto lo logra a traves de ciclo while y comparaciones para establecerse en la posición adecuada de escritura, tanto para escribir en la posición adecuada, como escribir la prioridad.

Notas

- El archivo .h está muy bien explicado y contiene todo lo necesario para poder trabajar, con los registros del NVIC.

- El archivo.c contiene una única función que permite poder habilitar una interrupción de forma general, sin embargo, para lograr que este sea un driver eficiente, hace falta la descripción de otras funciones que ayuden a poder acceder a los otros registros del NVIC.  
Ejemplo de la necesidad de estas demás funciones, es que no tenemos ninguna función que permita poder deshabilitar una interrupción si nosotros queremos, o también, no tenemos ninguna que nos pueda ayudar a ver las banderas de interrupción en caso que se necesiten, entre otras que hacen falta.