

## Dispositivos utilizados:

Leds: D3, D5, D7, D8 y D9.  
Pulsadores: S3, S4, S5 y S6.

## Sincronización de los dispositivos (encuesta/interrupción):

Pulsador S3: se gestiona por encuesta.  
Pulsador S4: se gestiona por interrupción.  
Pulsador S5: se gestiona por interrupción.  
Pulsador S6: se gestiona por interrupción.

## Funciones que llama el programa principal (nombre y funcionamiento)

**inic\_leds** (en GPIO.c)

- Inicializa los pines de todos los leds como salida (TRIS=0).

**inic\_pulsadores** (en GPIO.c)

- Define los pines analógicos AN16-AN31 como digitales. El pin RA7 (pulsador S5/led D10) es el AN23: hay que definirlo como digital debido a que se usa el pulsador S5.
- Inicializa los pines de los pulsadores S3, S4, S5 y S6 como entrada (TRIS=1).

**inic\_CN** (en CN.c)

- Inicializa el módulo CN para que los pulsadores S4 (pin CN19), S5 (pin CN23) y S6 (pin CN16) se gestionen por interrupción.

## Otras funciones importantes (no se les llama desde el main)

En este caso no hay más funciones

## Rutinas de atención

**Rutina módulo CN** (en CN.c): `_CNInterrupt()`

- Comprueba los pulsadores S4, S5 y S6 (los habilitados para interrumpir) y si están pulsados se conmuta el led correspondiente:
  - Pulsador S4 (RD13) -> led D3 (RA0)
  - Pulsador S5 (RA7) -> led D5 (RA2)
  - Pulsador S6 (RD7) -> led D7 (RA4)

## Funcionamiento general (verificado en las pruebas)

El programa principal realiza las inicializaciones de leds y pulsadores mediante las funciones descritas anteriormente (en ese orden). Después, se enciende el led D9 y se espera activamente hasta que se pulse S3 (gestión por encuesta). Una vez se pulsa S3, se apaga el led D9 y se enciende el led D8. Tras esto, se realiza la inicialización del módulo CN mediante la función descrita anteriormente. Tras la ejecución de la función *inic\_CN* (y no antes), el tratamiento de los pulsadores S4, S5 y S6 se realizará por interrupción. A continuación, se dará un bucle infinito durante el cual se tratarán las interrupciones que puedan darse mediante los pulsadores mencionados. De este modo, en cualquier momento en que se pulse uno de dichos pulsadores se conmutará el led correspondiente (ver `_CNInterrupt()`).

### **Conclusiones obtenidas en relación con el código escrito y las pruebas relacionadas**

En las pruebas de ejecución, se ha comprobado la existencia de los rebotes en los pulsadores S4, S5 y S6. Es decir, a veces la pulsación de dichos pulsadores se detecta más de una vez, y por tanto se conmuta el led correspondiente más de una vez, con lo que es posible que parezca que no ha cambiado de valor (cantidad par de cambios en la señal).

A raíz del hecho mencionado, hemos realizado pruebas a través del debugger y variables globales para ver el número de veces que se ha dado un cambio en la señal correspondiente a cada pulsador tratado por interrupción por cada pulsación. Para ello, hemos inicializado tres variables globales a 0, una por cada pulsador tratado por interrupción. Además, hemos modificado dichas variables en la rutina de *CN*, de modo que cada vez que se realice una pulsación de los pulsadores tratados por interrupción se incremente en 1 la variable correspondiente. De este modo, hemos visualizado el efecto de dichos rebotes observando los valores de dichas variables tras cada pulsación mediante el debugger. Por ejemplo, pulsando 5 veces el pulsador S6, hemos visto que en las primeras 4 pulsaciones se ha dado un único cambio en la señal mientras que en la quinta pulsación se han dado dos. Cabe destacar que al conmutar el led un número par de veces parecía que no se había alterado.