GUÍA DE EJERCICIOS N° 4
DRIVERS

Objetivo: Implementar sus propios *drivers*.

1. Timer
   - Implementar un *driver* de manejo de múltiples *timers* llamado `timer`, basado en el archivo `timer.c` que utiliza el módulo `SysTick` de la guía anterior.
   - Dicho *driver* debe utilizar como base de tiempo el módulo `SysTick`.
   - Modificar el proyecto `Baliza` para que utilice este módulo como base de tiempo, usando un *timer* para la lectura del pulsador y otro para el parpadeo de la baliza.

2. Encoder
   - Implementar un *driver* de manejo del encoder rotativo llamado `encoder`.
   - Se deben poder leer los pasos dados, siendo positivo si fueron en sentido horario o negativo en sentido antihorario.
   - Se debe realizar utilizando los módulos `gpio` y `timer/SysTick`.
   - Realizar un programa de prueba que cambie el color de LED RGB en una secuencia cíclica.

3. Button
   - Implementar un *driver* de manejo de botones llamado `button` que permitir leer el estado y eventos de varios botones (pulsadores), realizando antirrebote por software.
   - Se deben poder detectar eventos de PRESS, RELEASE, LKP (long-key press, mantener presionado el botón por un tiempo prolongado) y/o TYPEMATIC (muchos eventos de PRESS si se mantiene presionado un pulsador, similar a mantener presionada una tecla de la PC).

Problemas suplementarios

4. LEDs
   - Implementar un *driver* de manejo de LEDs llamado `led` que maneje distintos tipos de destellos de LEDs. Por ejemplo: titilar a velocidad rápida, titilar 3 veces a velocidad lenta
   - Los servicios del driver son a elección del alumno.
   - La implementación de los servicios no debe ser bloqueante.

5. Baliza PRO
   - Reescribir el programa Baliza para que funcione utilizando una correcta arquitectura de firmware.
   - Debe utilizar el *driver* `button` para la lectura del pulsador.
   - Debe utilizar el *driver* `led` para controlar el parpadeo de la baliza y el LED de estado.
   - La APP no debe incluir ni invocar a los módulos `gpio`, `SysTick` y `board`.