

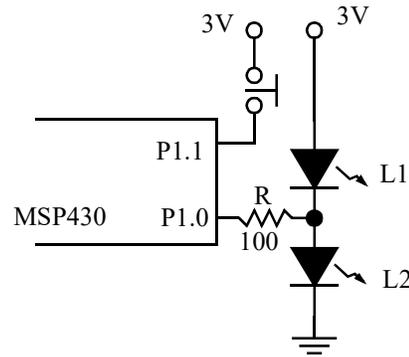
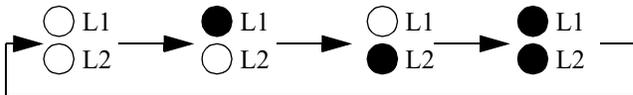
Apellidos:.....

Nombre:..... Puesto:.....

P1	P2

**Duración 2:00 horas**

1.- (5 PUNTOS) Considere el circuito de la figura. Haga un programa en ensamblador del MSP430 **basado en multitarea cooperativa** que inicialmente tiene los dos leds apagados. Cada vez que se acciona el pulsador, se cambian los leds encendidos según el ciclo:



Tenga en cuenta que no debe notarse cambio en el brillo aparente de los leds encendidos cuando se cambia de configuración.

Notas:

- Nótese que se pide un programa que use multitarea cooperativa, no que lea las teclas por interrupciones ni use el TimerA.
  - Minimice el consumo del sistema manteniéndolo todo el tiempo posible en modo de bajo consumo.
  - Considere el pulsador libre de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada y LFXT en marcha con cristal de 32768Hz.
  - Dispone del módulo `st.asm` y la subrutina `cmp32`.
- 2.- (5 PUNTOS) Considere el circuito de la figura. Se trata de un sistema de luces con encendido y apagado retardado. Cada vez que se pulsa el switch, se conmuta el estado del led. Pero el encendido/apagado no es instantáneo, sino que se hace variando suavemente el brillo. Haga un programa de mínimo consumo en ensamblador del MSP430 que gestione con el TA0 y por interrupciones la conmutación del estado del led cada vez que se pulse el switch.

Notas:

- La pendiente es de 100%/3s. El apagado/encendido siempre se empezará desde el nivel de brillo actual del led y con la pendiente inversa (vea figura). Use una frecuencia de 128 Hz y 32 niveles de brillo.
- Considere el pulsador libre de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada y LFXT en marcha con cristal de 32768Hz.
- La función primaria de P1.0 es TA0.1 (CCI1A para capturas y Out1 para salidas).
- La función primaria de P1.1 es TA0.2 (CCI2A para capturas y Out2 para salidas).

