

Apellidos:.....

P1	P2	P3

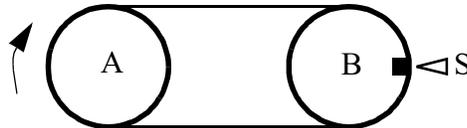
Nombre:..... Puesto:.....

Duración 4:00 horas

1.- [3 Puntos] Analice el siguiente código e indique qué hace. Proponga un código alternativo que haga lo mismo, de forma más eficiente. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/salida, ubicación y tipo. Calcule el número de ciclos de ejecución, así como el tiempo total para $f_{MCLK}=1MHz$. Ensamble manualmente el programa y dé en hexadecimal el código máquina.

		Emulación	Ciclos	Código máquina
;				
;				
hb	mov.w r12, r13	;		
	mov.w #0x8000, r12	;		
hbb1	rra.w r13	;		
	jc hbs	;		
	rrc.w r12	;		
	jnc hbb1	;		
	jmp hbf	;		
hbb2	rra.w r13	;		
	xor.w #1, sr	;		
hbs	rrc.w r12	;		
	jnc hbb2	;		
hbf	ret	;		

2.- [3 Puntos] En el mecanismo de la figura, las ruedas A y B tienen un perímetro de 50cm y están conectadas por una correa que las hace mover de forma solidaria. La rueda A es movida por un motor que se activa con la señal M. La B dispone de un sensor S que permite contar el número de vueltas de la misma (suponga que inicialmente B está posicionado de forma que S=1; conforme B gira, S se desactivará y se volverá a activar después de hacer un giro completo; vea la figura). Además, hay un pulsador con salida P que puede ser accionado por el usuario.



Se desea realizar un controlador para el mecanismo anterior cuya funcionalidad sea la siguiente: cuando el usuario pulsa P, el controlador activa el motor para que la cinta se mueva 1m controlando el estado de S. Una vez completado el movimiento, el controlador parará el motor incluso si P=1 y se asegurará de que P=0 antes de iniciar otro posible ciclo con una nueva pulsación de P.

Realice un programa en ensamblador usando la técnica de multitarea cooperativa. Dispone del módulo st.asm y la subrutina cmp32. Considere perro guardián desactivado, los puertos desbloqueados, pila inicializada y LFXT en marcha con cristal de 32768Hz.

3.- [4 Puntos] Considere el circuito de la figura. Se trata de un sistema de luces con encendido/apagado y control de intensidad en una sola tecla con rebotes. Cada vez que se pulsa momentaneamente el switch, se conmuta el estado del led. Inicialmente el led se enciende con un 100% de su capacidad. Si se actúa sobre el switch al menos un cuarto de segundo, se entra en el modo de cambio de intensidad hasta que es liberado. La misma varía entre el 25% y el 100% a un ritmo del 100%/2s. Cada vez que se llega a los límites, la variación de brillo cambia de sentido. En el momento de liberación del switch se paraliza el cambio de intensidad de la luz y se memoriza para posteriores encendidos/apagados.

Haga un programa de mínimo consumo en ensamblador del MSP430 que gestione con el TA0 y por interrupciones este comportamiento.

Notas:

- Use una frecuencia de 128 Hz y 32 niveles de brillo.
- El pulsador tiene un tiempo de rebote de 10ms.
- Si la pulsación es menor al tiempo de rebote, se ignorará la misma. Si el tiempo de pulsación es inferior a 0'25s, se considerará una pulsación momentánea y se procederá a conmutar el estado de la luz. Si es mayor, se entra en modo de cambio de intensidad (si el led estaba apagado, se enciende para proceder al ajuste de la intensidad).
- Considere 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).

