SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2024-2025**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

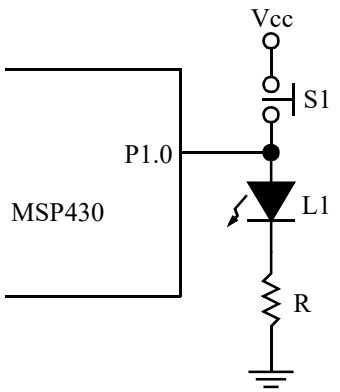
1.- Analice el siguiente código e indique qué hace. Proponga un código alternativo que haga lo mismo, pero de forma más eficiente. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.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

```
;                                   Emulación        Ciclos Código máquina
;-------------------------------------------------------------------------------
K           .equ    0x8000          ;_______________|   |_____________________
misterio:   mov.w   #K, r11         ;_______________|   |_____________________
misterioB:  rrc.w   r12             ;_______________|   |_____________________
            rrc.w   r13             ;_______________|   |_____________________
            rrc.w   r11             ;_______________|   |_____________________
            rrc.w   r13             ;_______________|   |_____________________
            rrc.w   r12             ;_______________|   |_____________________
            rrc.w   r11             ;_______________|   |_____________________
            jnc     misterioB       ;_______________|   |_____________________
            mov.w   r11, r12        ;_______________|   |_____________________
            ret                     ;_______________|   |_____________________
```

2.- Considere el circuito de la figura. Haga un programa en ensamblador del MSP430 ***basado en multitarea cooperativa*** que inicialmente configura el puerto como entrada para poder leer la tecla. Cuando el usuario la pulsa, el led se enciende sin intervención del MSP. El objetivo del programa es alargar el encendido del led para asegurar que el mismo está encendido un segundo más que el tiempo que ha estado pulsada. Para ello, cuando se suelta, se pone el puerto en modo salida para encender el led. Después de un segundo, el puerto se vuelve a poner como entrada, apagándose el led (si el pulsador no está accionado). Para evitar observar el apagado momentaneo del led, ejecute la tarea con un periodo máximo de 10ms.

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 función `cmp32`.



3.- Se desea hacer un transmisor que usa el código Morse (ver tabla). La duración del punto es T = 0’25s. Una raya tiene una duración de 3T. Entre cada par de símbolos de una misma letra

existe una ausencia de señal con duración T. Entre las letras de una misma palabra, la ausencia es de tres puntos (3T). Para la separación de palabras transmitidas el tiempo es de tres veces el de la raya (9T). En la tabla siguiente se recogen los códigos internacionales para las letras y los dígitos. Existen algunos símbolos más que no vamos a tener en cuenta:

A	.-	G	--.	M	--	S	...	Y	-.--	4	....-
B	-...	H	....	N	-.	T	-	Z	--..	5	.....
C	-.-.	I	..	O	---	U	..-	0	-----	6	-....
D	-..	J	.---	P	.--.	V	...-	1	.----	7	--...
E	.	K	-.-	Q	--.-	W	.--	2	..---	8	---..
F	..-.	L	.-..	R	.-.	X	-..-	3	...--	9	----.

Codificación en binario de los códigos Morse: se sustituyen los puntos por 0 y las rayas por 1. Sea N la longitud del código. Se añadirán 7-N '0' y un '1' por la izquierda. Por ejemplo, para la letra C, el código es "-.-." lo que se traduce en "1010" de longitud 4. Se añade un "0001" por la izquierda, quedando "00011010". De esta forma se codifica la longitud de la secuencia Morse y la propia secuencia en 8 bits. El proceso de descodificación consistirá en desplazar el código a la izquierda hasta que salga el primer 1. 8 menos el número de desplazamientos es la longitud de la secuencia, que ha quedado justificada a la izquierda en el byte desplazado. A continuación se muestran algunos ejemplos más:

```
TabMrsLetra.byte   0b00000101      ; A .-
            .byte   0b00011000      ; B -...
            .byte   0b00011010      ; C -.-.
```

La transmisión se hará por el puerto P1.0 que tiene conectado un led con lógica positiva. La función primaria del puerto está asociada a TA0.1. Implemente el módulo `morse.asm` que recoge los siguientes servicios con el mínimo consumo posible:

a) `void morseIni (void)`. Inicializa las variables del módulo, configurando el puerto de salida y arrancando el TA0.

b) `int morseTx (char c)`. Transmite el carácter `c` cuyo código ASCII se pasa. Si el carácter es un espacio, emite un silencio interpalabra. Si es una letra o un dígito, primero obtiene el código morse asociado y transmite el primer símbolo del mismo. El resto se hace por interrupciones. Devuelve 0 si todo fue bien, 1 si había una transmisión en marcha y 2 si el carácter `c` no es correcto (no es una letra, espacio o dígito). No se distingue entre mayúsculas y minúsculas.

c) `ISR`. Subrutina de servicio de interrupciones del TA0. Transmite los símbolos morse salvo el primero que se hace en la función anterior. Añade los espacios intersímbolos e intercarácter y se desactiva cuando la tranmisión ha terminado.