

Apellidos:.....

P1	P2	P3

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

Duración 4:00 horas

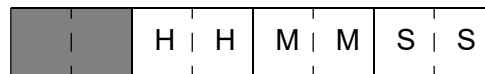
1.- [3 Puntos] Preguntas cortas:

- a) La instrucción NOP no hace nada, pero ocupa una palabra de código y tarda un ciclo en ejecutarse. Proponga una solución alternativa que ocupe una palabra y pierda 2 ciclos.
- b) Cuando se produce una interrupción, el procesador termina de ejecutar la instrucción en curso y, si las interrupciones están habilitadas, inicia el proceso de servir la más prioritaria de las pendientes. Por tanto, una instrucción en sí misma es atómica, ya que no se puede dividir. Sin embargo, un trozo de código cualquiera formado por más de una instrucción no es atómico. Indique cómo lo convertiría en atómico.
- c) Escriba las instrucciones necesarias para que la frecuencia del ACLK sea 1'17 kHz (NOTA: F1fxt=32768Hz, Fv1o=9400Hz, F1fmodclk=37500Hz).
- d) Haga un programa en ensamblador que tenga dos tareas cooperativas. La primera hace parpadear el led1 del Launchpad a 3 Hz (es decir, se apaga y se enciende 3 veces en un segundo) y la segunda hace lo propio con el led2 a 7Hz. Dispone del módulo st.asm desarrollado en prácticas. Escoja la mínima frecuencia del TA2 para que el error de las frecuencias esté por debajo del 0'1%. Led1 en P1.0; Led2 en P9.7.

2.- [3 Puntos] Se desea realizar la función IncTime que recibe una información de hora y la incrementa en un segundo:

```
int IncTime (unsigned long *hms);
```

hms es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):



La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba IncTime en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 Puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que alargue el encendido del led hasta 3s cuando la pulsación sea menor a ese tiempo. Para ello, se medirá el tiempo de la pulsación y, si es menor, colocará el puerto en modo salida y encenderá el led el tiempo restante hasta completar los 3s. Nótese que el led se apagará brevemente desde que se suelte la tecla hasta que se encienda por programa. Una vez terminado el tiempo, volverá a colocar el puerto en modo entrada para esperar la siguiente pulsación. En caso de que la pulsación fuera de 3s o más, no se hará ninguna actuación. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0

cuando es salida.

