

Fecha			Curso	Calificaciones Parciales			Cal. Final
1	7	2020	2				

Examen de laboratorio abreviado

Para la realización de este examen dispone de 55 minutos. Adjunto al enunciado encontrará en Moodle un fichero comprimido que, al descomprimirlo (dentro de la carpeta MICR de las prácticas), genera una estructura de carpetas (una por cada hito) en la que debe ir trabajando. Al finalizar el mismo deberá comprimir (en formato .7z, no incluya la biblioteca *mbed* ni las carpetas *~build* y *~listings*) la carpeta MICR/Ex y subirla al correspondiente enlace en Moodle.

Para cada hito verá una lista de objetivos que debe cumplir su programa. Si no logra todos ellos no se podrá obtener la máxima calificación.

Para la realización de este examen **no** se permite el uso de la función `wait()`, tampoco se permite el uso de la librería `sw_tick_serial`, pero no es necesario el empleo de máquinas de estados finitos (FSM) para resolver ninguno de los hitos. Sí puede reutilizar el código que haya escrito anteriormente durante las prácticas. No obstante lo anterior, **el código que presente para evaluar deberá estar libre de líneas sobrantes procedentes de código antiguo.**

Hito 1 (40 puntos): CONTADOR DE PULSACIONES

El sistema debe detectar pulsaciones en los pulsadores izquierdo y derecho, gestionando adecuadamente los rebotes. El sistema debe disponer de una variable **no global** llamada `pulsaciones_m` cuyo valor se incrementará en 1 con cada pulsación del pulsador izquierdo, y se decrementará en 1 con cada pulsación del pulsador derecho. Dicha variable no deberá tener nunca un valor negativo o superior a 99, las pulsaciones que hicieran sobrepasar estos límites deben ser ignoradas. El valor de la variable `pulsaciones_m` debe mostrarse utilizando los dos *displays* de 7 segmentos y su valor inicial será 50.

CRITERIOS:

Vº. Bº:

- ☐ Se llama a `wait()` o se incluye o *linka* `sw_tick_serial` (-40)
- ☐ Se aprecian rebotes o la detección de las pulsaciones no es correcta (-15)
- ☐ El valor mostrado en los *displays* no es correcto (-15)
- ☐ La multiplexación de los *displays* no es correcta (-15)
- ☐ La variable cuenta mal el número de pulsaciones (-15)
- ☐ La variable es global (-10)
- ☐ Hay código sobrante (-10)
- ☐ ISR complejas (-15)
- ☐ Gestión del sueño incorrecta (-10)
- ☐ Otros (a valorar):

Hito 2 (30 puntos): REPRESENTACIÓN LUMÍNICA

Modifique el código del hito anterior para que el comportamiento de los LED siga las siguientes instrucciones:

1) si el número mostrado en los *displays* de 7 segmentos estuviera entre 01 y 50, el LED izquierdo permanecerá encendido con una intensidad que variará linealmente con el número mostrado; siendo mínima para un valor de 1 y máxima para un valor de 50. Los LED central y derecho permanecerán apagados.

2) si el número mostrado en los *displays* de 7 segmentos estuviera entre 51 y 99, el LED izquierdo permanecerá apagado y el LED derecho permanecerá encendido, con una intensidad que variará linealmente con el número mostrado; siendo mínima para un valor de 51 y máxima para un valor de 99. El LED central permanecerá apagado.

3) si el número mostrado en los *displays* de 7 segmentos fuera 00, todos los LED permanecerán apagados.

Tenga en cuenta que para controlar la intensidad de los LED no se permite emplear la clase `PwmOut`; en su lugar establezca una frecuencia de parpadeo de los LED de 100 Hz. La restante funcionalidad del hito anterior debe conservarse inalterada.

CRITERIOS:

Vº. Bº:

- ☐ Se llama a `wait()` o se incluye o *linka* `sw_tick_serial` (-30)
- ☐ La multiplexación de los *displays* no es correcta, siéndolo en el hito anterior (-10)
- ☐ Los LED no funcionan como se especifica (-30)
- ☐ Se observa el parpadeo de los LED (-15)
- ☐ El resto de los pulsadores influyen en el funcionamiento (-10)
- ☐ Hay código sobrante (-10)
- ☐ ISR complejas, siendo correcto en el hito anterior (-10)
- ☐ Gestión del sueño incorrecta, siendo correcta en el hito anterior (-5)
- ☐ Otros (a valorar):

Hito 3 (30 puntos): MEDIDA DE LUZ AMBIENTE

Modifique el código del hito anterior para que al detectar una pulsación en el pulsador central el sistema haga una medida de la luz ambiente (0 = oscuridad total, 99 = iluminación máxima). El valor de dicha medida debe sustituir al valor de cuenta mostrado en los *displays* y la iluminación de los LED del Hito 2 debe ajustarse inmediatamente al valor leído. Este valor se podrá incrementar y decrementar tal y cómo se especifica en el primer Hito.

La restante funcionalidad de los hitos anteriores debe conservarse inalterada.

CRITERIOS:

Vº. Bº:

- ☐ Se llama a `wait()` o se incluye o *linka* `sw_tick_serial` (-30)
- ☐ La medida de luminosidad del sistema no es correcta o sólo se hace una vez (-15)
- ☐ La medida de luminosidad no se muestra correctamente en los *displays* (-15)
- ☐ Tras una medida, el valor no puede modificarse con los pulsadores (-15)
- ☐ Tras una medida, la representación lumínica de los LED no se ajusta al valor (-15)
- ☐ La restante funcionalidad de los hitos anteriores se ha modificado o degradado (-10)
- ☐ Hay código sobrante (-10)
- ☐ ISR complejas, siendo correcto en los hitos anteriores (-10)
- ☐ Gestión del sueño incorrecta, siendo correcta en los hitos anteriores (-5)
- ☐ Otros (a valorar):