

 <b>UNIVERSIDAD</b>  <b>POLITÉCNICA</b>  <b>DE MADRID</b>  <b>ETSIS Telecomunicación</b>			<b>APELLIDOS:</b>				
			<b>NOMBRE:</b>			<b>DNI:</b>	
			<b>MICROPROCESADORES</b>				
			<b>Grupo Lab.:</b>			<b>Puesto lab.:</b>	
<b>Fecha</b>			<b>Curso</b>	<b>Calificaciones Parciales</b>			<b>Cal. Final</b>
<b>20</b>	<b>11</b>	<b>2018</b>	<b>2</b>				

Para la realización de este examen dispone de 50 minutos. Debe crear en el ESCRITORIO de su PC una carpeta por cada hito, cuyo nombre debe tener el formato APELLIDOS\_nombre\_Hito\_X (con X de 1 a 3). Al finalizar el examen debe comprimir TODAS las carpetas en un único fichero 7-ZIP y subirlo 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. Cuando considere que tiene uno de los hitos funcionando debe levantar la mano para que el profesor pueda ir a validarlo. El profesor evaluará cada hito a cada alumno UNA ÚNICA VEZ. Una vez evaluado un hito no se puede modificar el código evaluado y debe pasar al siguiente hito. En cualquier caso NO SE QUEDE ESPERANDO A QUE LLEGUE EL PROFESOR, continúe con el siguiente hito (EN OTRA CARPETA). El profesor revisará todos los hitos pendientes.

Para la realización de este examen **no** se permite el uso de la función `wait()`. Debe descargar de *Moodle* el proyecto de *Keil µVision 5* disponible para su plataforma (LPC1768 o STM32L432KC) del enlace en la zona de este examen. Dicho proyecto incorpora las librerías *sw\_tick\_serial* y *to\_7seg* (además de *mbed*) aunque el fichero *main.c* incluido está muy incompleto (de hecho, ni siquiera se compila) y debe completarlo para alcanzar la funcionalidad pedida para cada hito. No se permite la utilización de ningún recurso *software* ajeno a lo disponible en el *Moodle* de la asignatura, en el que sus entregas de las anteriores prácticas no están accesibles. No se permite el uso de *pen-drives*, discos USB o cualquier otro medio para el almacenamiento de datos.

### Hito 1 (35 puntos): DECREMENTO

El sistema debe emplear el pulsador derecho. Inicialmente (tras un *reset*) en el *display* de la derecha se mostrará el dígito «9». Dicho dígito se decrementará con cada pulsación del pulsador derecho. Al llegar a cero, una nueva pulsación volverá a poner el *display* a 9. A la vez de todo lo anterior, el LED derecho lucirá de forma intermitente a una frecuencia de 0.5 Hz (1 s encendido, otro segundo apagado). Los LED restantes, así como el *display* izquierdo, permanecerán apagados en todo momento. Los demás pulsadores no afectarán al funcionamiento.

El sistema debe, además, contar el número de veces que se pulsa el pulsador derecho en una variable **global** (que será del tipo adecuado para contener un número de 16 bits sin signo) llamada *cnt\_switch*. Cuando muestre este hito al profesor este le pedirá que active un punto de ruptura (*breakpoint*) que permita conocer el valor de dicha variable.

#### CRITERIOS:

Vº. Bº:

- ☐ Se llama a `wait()` (-35)
- ☐ Los mensajes mostrados en el *display* no son los indicados (-15)
- ☐ La respuesta al pulsador derecho no es la esperada (-15)
- ☐ El funcionamiento del LED derecho no es correcto (-10)
- ☐ No sabe poner un punto de ruptura (-15)
- ☐ No sabe ver el valor de una variable (-15)
- ☐ La variable cuenta mal las veces que se pulsa el pulsador derecho (-10)
- ☐ La variable no es del tipo adecuado o no es global (-10)
- ☐ Los demás pulsadores influyen en el funcionamiento o los demás LED o el *display* izquierdo se encienden (-10)
- ☐ El brillo del *display* varía en el tiempo o se aprecian parpadeos (-10)
- ☐ Otros (a valorar):

## Hito 2 (35 puntos): MULTIPLEXACIÓN

Modifique el código del hito anterior para que la cuenta descendente mostrada en el *display* empiece en 99 y se muestren dos cifras. Cuando enseñe este hito al profesor, este le pedirá que, empleando los recursos de depuración de la herramienta (sin alterar el programa) modifique el valor de la cuenta del *display*, de modo que se represente «03» sin necesidad de realizar las 96 pulsaciones que serían necesarias para conseguirlo y que, de esta forma, sea cómodo verificar que tras el estado «00» se pasa al «99». La restante funcionalidad del hito anterior debe permanecer inalterada.

### CRITERIOS:

Vº. Bº:

- ☐ Se llama a `wait()` (-35)
- ☐ El valor mostrado en el *display* no es correcto (-15)
- ☐ La multiplexación de los *displays* no es correcta o se aprecian sombras (-15)
- ☐ No sabe modificar el valor de la cuenta para forzar la visualización del «03» (-15)
- ☐ Tras el «00» no se pasa al «99» (-10)
- ☐ La restante funcionalidad del hito 1 se ha modificado o degradado (-15)
- ☐ Otros (a valorar):

## Hito 3 (30 puntos): PRINTF()

Modifique el código del hito anterior para que, además, cada 10 s, el valor de la cuenta se incremente automáticamente en 3 unidades. También, cada vez que pasen esos 10 s, se enviará hacia el PC —mediante `printf()`, para ser visualizado en *Tera-Term*— el mensaje, sin incluir las comillas: «Otros 10 s (x)», donde *x* es el valor de la cuenta y cada mensaje aparece en una línea nueva. La cuenta nunca se incrementará por encima de 99. La restante funcionalidad de los hitos anteriores debe conservarse inalterada. Se recuerdan ahora los valores a los que debe ajustarse, en *Tera-Term*, el puerto de comunicaciones cuando se usa la librería *sw\_tick\_serial*:

Speed:	115200 ▼
Data:	8 bit ▼
Parity:	none ▼
Stop bits:	1 bit ▼
Flow control:	none ▼

### CRITERIOS:

Vº. Bº:

- ☐ Se llama a `wait()` (-30)
- ☐ No ocurre el incremento automático de la cuenta (-15)
- ☐ El incremento ocurre a una frecuencia o en una cantidad distinta a las esperadas (-10)
- ☐ No sabe configurar adecuadamente *Tera-Term* (-15)
- ☐ No ocurriendo lo anterior, los datos visualizados en *Tera-Term* no son los esperados (-15)
- ☐ La restante funcionalidad de los hitos anteriores se ha modificado o degradado (-10)
- ☐ Otros (a valorar):