

 <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>01</b>	<b>2021</b>	<b>2</b>				

Para la realización de este examen dispone de 50 minutos. Descomprima el fichero descargable de Moodle, lo que dará lugar a la siguiente estructura de carpetas:

```

MICR
├── EX_L2
│   └── Hitos
│       ├── Hito_1
│       ├── Hito_2
│       └── Hito_3
├── mbed
└── to_7seg
  
```

Debe trabajar dentro de las carpetas Hito\_1 a Hito\_3, en la primera de ellas encontrará un proyecto de *Keil*, aunque su fichero `main.cpp` está muy incompleto. **Al finalizar el examen debe comprimir la carpeta Hitos** (borrando antes las carpetas `~build` y `~listings` que pudieran existir) 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 listo debe levantar la mano para que el profesor pueda validarlo. El profesor evaluará cada hito a cada estudiante 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. El profesor revisará todos los hitos pendientes.

Para la realización de este examen **no** se permite el uso de la función `wait()` ni de la librería `sw_tick_serial`. Los programas que realice deben «dormir» al procesador siempre que sea posible. 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): LED INTERMITENTE

El sistema debe encender de forma intermitente el LED derecho. La frecuencia de la intermitencia será de 2 Hz y el ciclo de trabajo del 80%. Note que el pin conectado a dicho LED no posee funcionalidad PWM en ninguna de las placas mbed LPC1768 o Nucleo-l432kc. El resto de LED y el *display* deben permanecer apagados. Debe dormirse al procesador cuando sea posible.

El sistema debe, además, contar el número de veces que se *enciende* el LED en una variable **local** (que será del tipo adecuado para contener un número de 8 bits sin signo) llamada `cnt_led`. Esta variable se incrementará en el instante en que el LED pase de *apagado a encendido*. Cuando muestre este hito al profesor, este le pedirá que, mediante el uso de un punto de ruptura (*breakpoint*), detenga la ejecución del programa cuando se *apague* el LED para, de esta forma, conocer el valor de dicha variable.

#### CRITERIOS:

Vº. Bº:

- ☐ Se llama a `wait()` o se usa `sw_tick_serial` (-35)
- ☐ El LED derecho no parpadea o lo hace de forma errática (-35)
- ☐ La frecuencia o el ciclo de trabajo del LED derecho no son los indicados (-15)
- ☐ 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 enciende el LED (-10)
- La variable no es del tipo adecuado o no es local (-10)
- El código dentro de las ISR es complejo (-10)
- No se duerme al procesador cuando es posible o se hace incorrectamente (-10)
- Los pulsadores influyen en el funcionamiento o los demás LED o cualquier *display* se encienden (-10)
- Otros (a valorar):

## Hito 2 (35 puntos): MULTIPLEXACIÓN

A la funcionalidad del hito anterior añada (en la carpeta Hito\_2) el que en el *display* de 7 segmentos se muestre, en decimal con dos cifras, el valor de la variable `cnt_led`. La cuenta se incrementará en coincidencia con el instante en el que el LED pasa de apagado a encendido. Tras el «99», `cnt_led` pasará a «00» y seguirá contando. Debe dormirse al procesador cuando sea posible.

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 `cnt_led`, de modo que se represente «95» sin necesidad de esperar cerca de 1 minuto para que, de esta forma, sea cómodo verificar que tras el «99» se pasa al «00». La restante funcionalidad del hito anterior debe permanecer inalterada.

### CRITERIOS:

Vº. Bº:

- Se llama a `wait()` o se usa `sw_tick_serial` (-35)
- No se ha implementado la multiplexación de los *displays* o está inoperativa (-35)
- La multiplexación de los *displays* no es correcta o se aprecian sombras (-15)
- El valor mostrado en el *display* no es correcto (-15)
- La cuenta es correcta pero no se actualiza en coincidencia con el encendido del LED (-10)
- El brillo del *display* varía (-15)
- No sabe modificar el valor de `cnt_led` para forzar la visualización del «95» (-15)
- Tras el «99» no se pasa al «00» o la cuenta se detiene (-10)
- La restante funcionalidad del hito 1 se ha modificado o degradado (-15)
- El código dentro de las ISR es complejo (-5)
- No se duerme al procesador cuando es posible o se hace incorrectamente (y en el hito anterior la gestión era correcta) (-10)
- Los pulsadores influyen en el funcionamiento o los demás LED se encienden (-10)
- Otros (a valorar):

## Hito 3 (30 puntos): PULSADOR

Modifique (en la carpeta Hito\_3) el código del hito anterior para que, además, `cnt_led` se incremente en 7 unidades cada vez que se actúe sobre el pulsador derecho. Deben tratarse convenientemente los rebotes en el pulsador. Si `cnt_led` fuese superior a 92 y se activase el pulsador, se restarían 100 unidades antes de sumar las 7 correspondientes a la pulsación, de modo que siempre sea  $0 \leq \text{cnt\_led} \leq 99$ . Debe dormirse al procesador cuando sea posible. La restante funcionalidad de los hitos anteriores debe conservarse inalterada.

### CRITERIOS:

Vº. Bº:

- Se llama a `wait()` o se usa `sw_tick_serial` (-30)
- No ocurre el decremento de la cuenta al actuar sobre el pulsador (-30)
- Se aprecian rebotes, ya sea tanto al pulsar como al soltar, o se no se registran todas las pulsaciones (-15)
- `cnt_led` se incrementa en 7 unidades en un instante diferente a la pulsación del botón, sin ser atribuible este comportamiento a los rebotes (-10)
- La restante funcionalidad de los hitos anteriores se ha modificado o degradado (-10)
- No se duerme al procesador cuando es posible o se hace incorrectamente (y en los hitos anteriores la gestión era correcta) (-5)
- Los restantes pulsadores influyen en el funcionamiento o los demás LED se encienden (-10)
- Otros (a valorar):