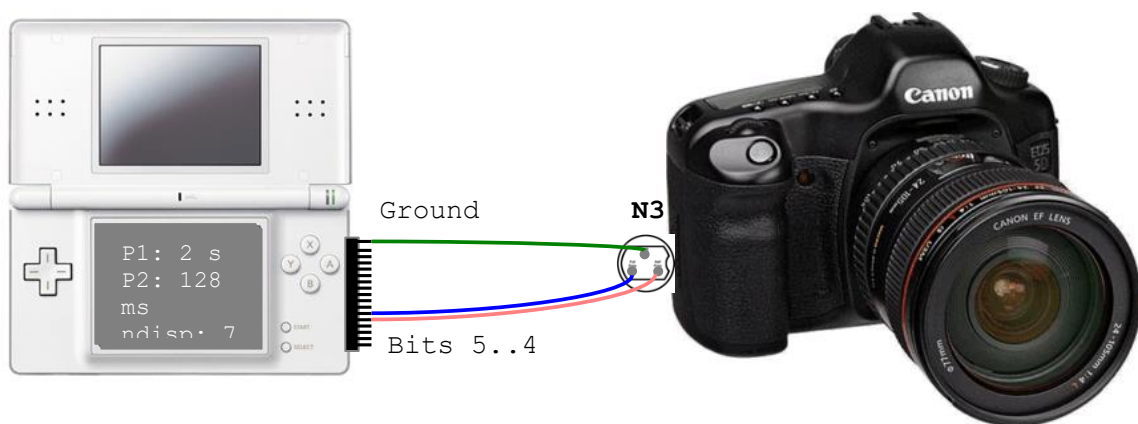


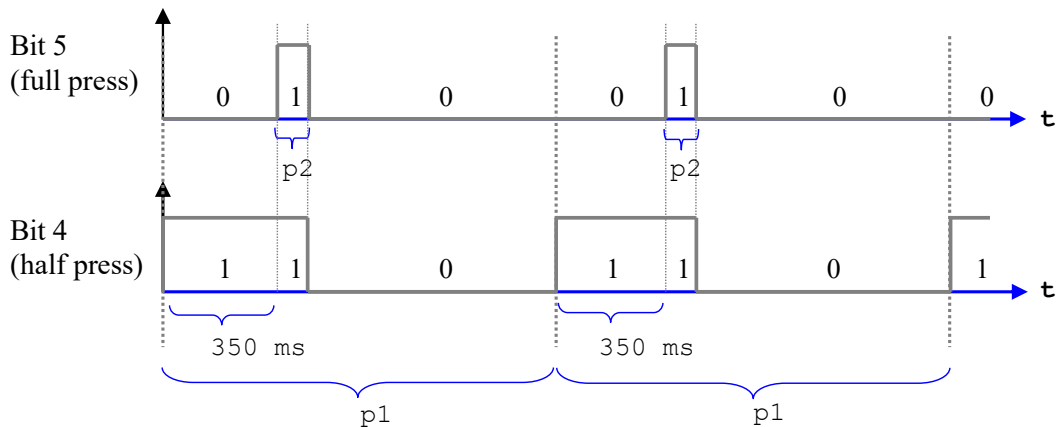
Problema 25: Disparador de cámara réflex (Ex. 2ª Conv. 2017-18)

Hay que controlar el botón de disparo de una cámara fotográfica réflex a través de un conector N3 de Canon®. Este conector dispone de 3 pines, uno de masa eléctrica (*Ground*), otro que activa la función de media presión del botón de disparo (*half press*) y otro que activa la función de presión completa (*full press*). Se utilizará una NDS con una interfaz de Entrada/Salida de 16 bits, que conectará los bits 4 y 5 de un registro de nombre simbólico REG_SHUTTER a los pines de *half press* y *full press* (respectivamente) de una cámara con conector N3 o compatible. El resto de bits del registro de E/S no se utilizan. A continuación se muestra un esquema de la conexión entre la NDS y un dispositivo de ejemplo:



La función de *half press* indica a la cámara que enfoque y que calcule los ajustes de exposición según la luz entrante (apertura del diafragma, tiempo de exposición, sensibilidad del sensor de imagen, etc.). Se supone que cualquier cámara profesional tardará menos de 350 milisegundos en realizar dichas tareas. La función de *full press* da la orden de realizar la foto con los ajustes obtenidos. Además, se puede controlar el tiempo de exposición mediante el tiempo durante el cual el pin de *full press* esté a 1.

El programa a realizar enviará un 1 por el bit 4 durante 350 milisegundos. Después enviará un 1 por los bits 4 y 5 durante un determinado periodo de exposición especificado por el usuario (**p2**). Además, el usuario también podrá indicar al programa que realice un cierto número de fotos consecutivas, con otro determinado intervalo de tiempo entre foto y foto (**p1**). El siguiente esquema muestra un ejemplo de la forma de onda de los bits 4 y 5 para dos fotos consecutivas:



Se dispone de las siguientes rutinas, ya implementadas:

<i>Rutina</i>	<i>Descripción</i>
<code>inicializaciones()</code>	Inicializa el <i>hardware</i> (pantalla, interrupciones, etc.)
<code>tareas_independientes()</code>	Tareas que no dependen de la gestión de los disparos (ej. medición de la cantidad de luz recibida por el sensor); t. ejecución ≈ 50 ms
<code>activar_timers01(int period)</code>	Activa los <i>timers</i> 0 y 1 en modo encadenado, para generar la activación de la IRQ del <i>timer</i> 1 cada cierto tiempo, especificado por parámetro en segundos
<code>activar_timer2(int period)</code>	Activa el <i>timer</i> 2 para generar la activación de su IRQ cada cierto tiempo, especificado por parámetro en milisegundos
<code>desactivar_timers(int patrnr)</code>	Desactiva el funcionamiento de los <i>timers</i> indicados por parámetro mediante un patrón de cuatro bits (del bit 3 al bit 0), donde cada bit provocará la desactivación (1) o no (0) del <i>timer</i> correspondiente
<code>scanKeys()</code>	Captura el estado actual de los botones de la NDS
<code>int keysDown()</code>	Devuelve un patrón de bits con los botones pulsados
<code>swiWaitForVBlank()</code>	Espera hasta el próximo retroceso vertical
<code>printf(char *format,...)</code>	Escribe un mensaje en la pantalla inferior NDS

Se propone usar las siguientes variables globales:

```
unsigned char num_disp = 0;           // número de disparos pendientes
unsigned char ind_p1 = 0;             // índice del periodo p1
unsigned char ind_p2 = 0;             // índice del periodo p2
unsigned char status = 0;              // estado actual (0 → parado,
                                     // 1 → half press,
                                     // 2 → half+full press)
unsigned char actualizar_info = 0;     // indica si hay que actualizar
                                     // información en pantalla
```

La variable `num_disp` indicará el número de disparos pendientes. El usuario podrá incrementar dicho número pulsando el botón `KEY_START`.

Las variables `ind_p1` e `ind_p2` almacenarán un índice entre 0 y 9, que permitirá obtener rápidamente el valor de los periodos **p1** y **p2** como 2 elevado al índice correspondiente. Se sugiere esta codificación porque proporciona un amplio rango de niveles con pocos valores. Concretamente, para el periodo de exposición **p2** obtendremos diez valores entre 1 y 512 milisegundos, lo cual se aproxima a los tiempos de exposición utilizados habitualmente por las cámaras fotográficas (1/1000, 1/500, 1/250, 1/125, 1/60, 1/30, 1/15, 1/8, 1/4, 1/2 segundos). El usuario podrá cambiar el valor del índice `ind_p2` con los botones `KEY_UP` y `KEY_DOWN`. Por otro lado, el periodo de los disparos **p1** abarcará un rango entre 1 y 512 segundos. El usuario podrá cambiar el valor del índice `ind_p1` con los botones `KEY_LEFT` y `KEY_RIGHT`.

La variable `status` permitirá controlar en qué estado del ciclo de un disparo se encuentra el sistema, pasando por los valores 1 (*half press*), 2 (*half+full press*), 0 (reposo).

La variable `actualizar_info` sirve para indicar cuándo ha habido un cambio en alguno de los valores de configuración principales, esto es, en el número de disparos pendientes o en alguno de los dos periodos, ya sea por pulsación de botones o porque se ha generado un nuevo disparo automáticamente (mediante las RSIs). Cada vez que haya un cambio de información, por la pantalla inferior de la NDS habrá que visualizar los siguientes mensajes:

```
Periodo 1: p1 s
Periodo 2: p2 ms
Disparos pendientes: nd
```

donde *p1*, *p2* y *nd* se tienen que sustituir por su valor correspondiente.

Para gestionar la activación automática de disparos, se pide usar la RSI del *timer* 1 para generar un nuevo disparo si quedan disparos pendientes, cada vez que pase el tiempo correspondiente al periodo **p1**. Si no quedan más disparos, la propia RSI puede desactivar los *timers* 0 y 1 para detener el proceso automático.

Por otro lado, se pide usar la RSI del *timer 2* para controlar el estado de los bits de cada disparo, pasando de *half press* a *half+full press* después de 350 milisegundos desde el inicio de la activación, y de *half+full press* a reposo después de los milisegundos correspondientes al periodo **p2**, desactivando el *timer 2* en esta segunda transición.

Por último, hay que tener en cuenta que si el proceso de activación automática está detenido y el usuario pide un nuevo disparo, habrá que iniciar dicho proceso desde el programa principal. Además, el usuario podrá cambiar los valores de configuración (`num_disp`, `ind_p1`, `ind_p2`) en cualquier momento, incluso si el proceso de activación automática ya está en marcha.

Se pide:

Programa principal en C, RSIs del *timer 1* y del *timer 2* en ensamblador.