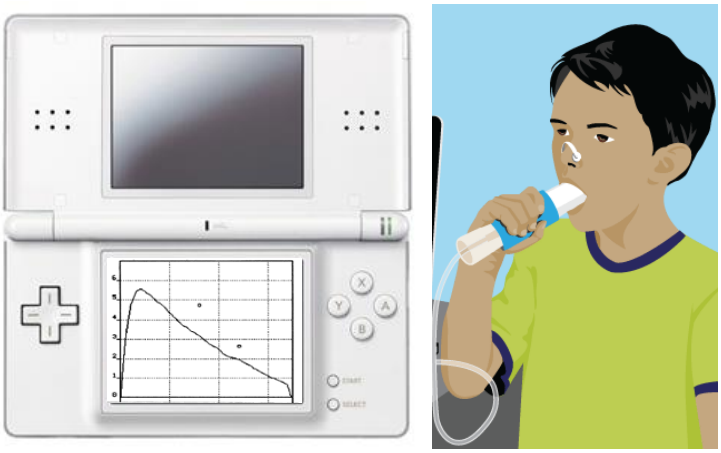
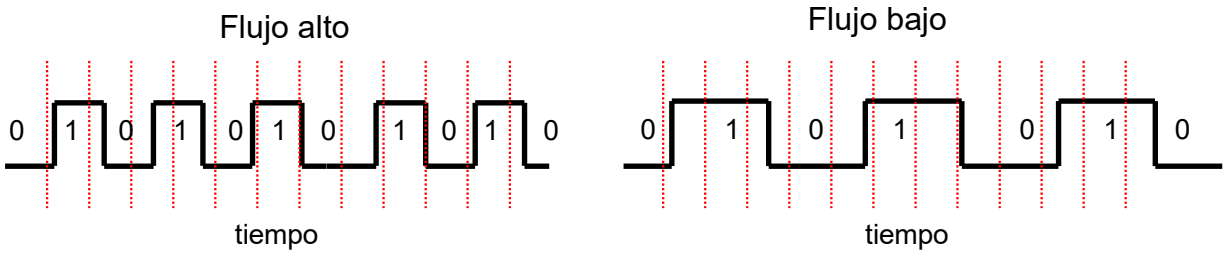
Problema 10: Espirómetro (Ex. 2ª Conv. 2011-12)

Se propone construir un espirómetro con una NDS. Un espirómetro es un aparato para medir la capacidad pulmonar. Consta de un tubo equipado con una hélice, la cual gira como consecuencia del aire que insufla el usuario:



La hélice genera un tren de impulsos (0/1/0/1/0/...) que indicará el nivel del flujo del aire; cuanto más alto sea el flujo, más impulsos (cambios de 0 a 1) por unidad de tiempo se obtendrán. Ejemplos:



El tren de impulsos se recibirá por el bit 0 del registro de E/S `0x0A000180` (reg. de 8 bits). El programa utilizará las interrupciones del *timer* 0 para consultar el valor de este bit periódicamente. Se dispone de una rutina ya implementada, de nombre `inicializar_timer0()`, que programa la interrupción IRQ_TIMER0 a una frecuencia aproximada de 2 KHz.

El programa esperará la pulsación de la tecla START para empezar a contar impulsos. Se supone que el flujo máximo de una persona normal puede generar entre 200 y 400 impulsos por segundo. Se puede suponer que nunca se llegará hasta los 600 impulsos por segundo.

Después de pulsar la tecla START, el programa representará gráficamente el nivel del flujo de aire de cada instante durante 10 segundos. Concretamente, hay que pintar un punto en la gráfica cada 5 centésimas de segundo. Esto supone un total de 200 puntos para toda la gráfica.

El valor del tiempo nos proporciona la coordenada X y el nivel de flujo nos proporciona la coordenada Y. Concretamente, hay que contar el número de impulsos para cada intervalo de tiempo, es decir, cada 5 centésimas. El número máximo de impulsos por intervalo es de 600/20, es decir, 30.

Se dispone de las siguientes rutinas, ya implementadas:

Rutina	*Descripción*
`inicializaciones()`	Inicializa pantalla e interrupciones
`scanKeys()`	Captura las teclas
`int keysDown()`	Devuelve el estado de las teclas pulsadas
`swiWaitForVBlank()`	Espera retroceso vertical
`dibujar_ejes()`	Dibuja los ejes para representar el gráfico
`add_pixel(int px, int py)`	Añade un píxel al gráfico, según las coordenadas de pantalla `px` (20-220) y `py` (0-180)
`actualizar_grafico()`	Actualiza el dibujo del gráfico

La rutina `dibujar_ejes()` solo se tiene que llamar una vez, antes de empezar la captura de los niveles de flujo. La rutina `add_pixel()` tarda menos de 100 microsegundos en ejecutarse. La rutina `actualizar_grafico()`, sin embargo, puede tardar hasta 5 milisegundos en ejecutarse. Esta rutina se asegura de activar todos los píxeles entre el último píxel añadido y el penúltimo, de modo que todo el gráfico sea una línea continua.

Se debe realizar una rutina que se encargará de convertir los valores de tiempo y flujo en coordenadas de pantalla, para que se puedan activar los píxeles correspondientes:

```
void convertir_punto(int ppx, int ppy);
```

A la llamada de la rutina, los valores de entrada (tiempo, flujo) se deben almacenar en R0 y R1. Al retorno de la llamada, los mismos registros contendrán las coordenadas en píxeles (`px`, `py`). Hay que tener en cuenta que el valor de tiempo se debe desplazar 20 píxeles a la derecha para ajustarse a la gráfica, mientras que el valor de flujo se debe multiplicar por 6 y restar de 180.

Se pide:

Programa principal en C, RSI del *timer* 0 y rutina `convertir_punto()` en ensamblador.