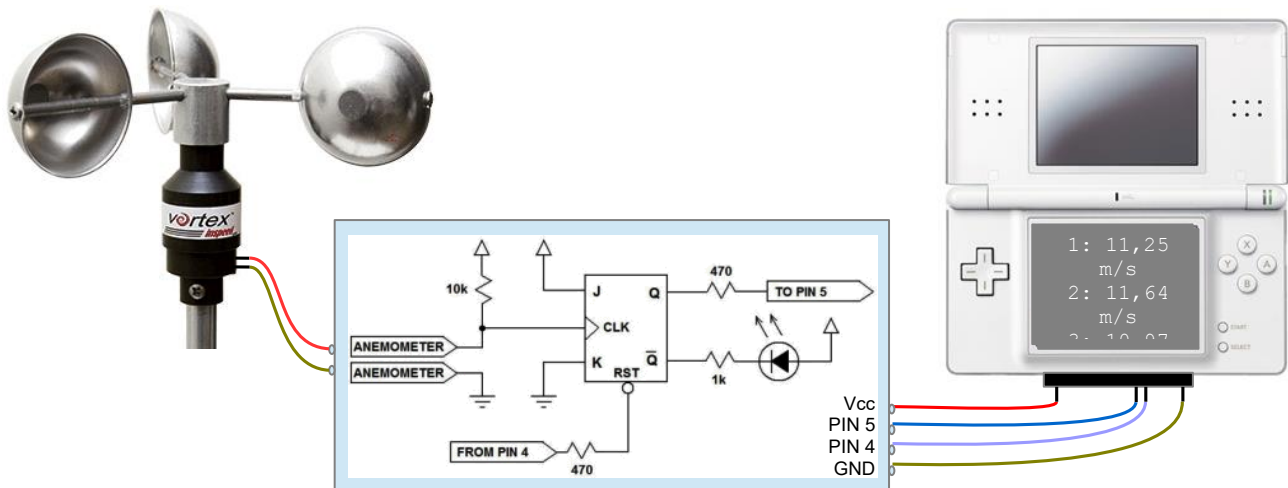


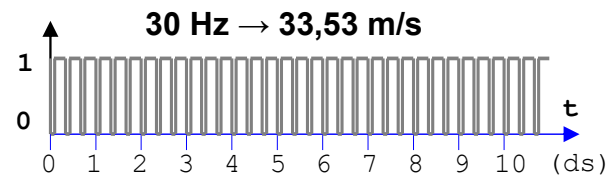
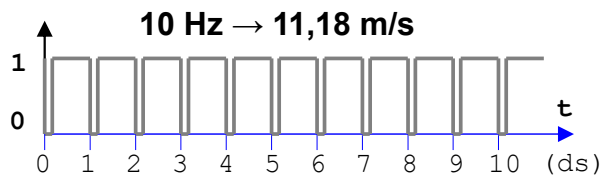
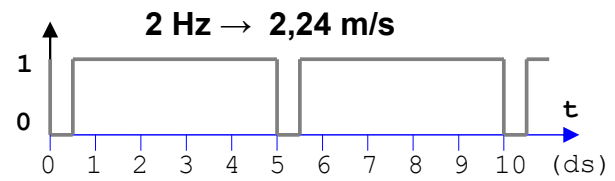
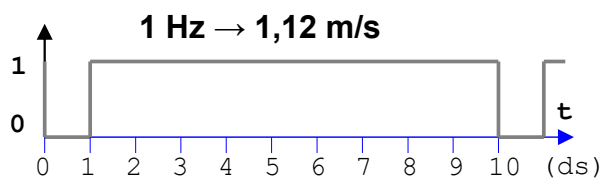
Problema 19: Anemómetro electrónico (Ex. 1ª Conv. 2016-17)

Se propone conectar un anemómetro electrónico *Vortex wind sensor* (Inspeed.com) a la NDS, con el fin de medir la velocidad del viento. El anemómetro genera un contacto eléctrico entre sus dos terminales cada vez que el rotor, impulsado por el viento, da una vuelta completa. Como el contacto dura un tiempo variable (en función de la velocidad angular), se ha añadido un circuito electrónico basado en un biestable J-K para convertir dichos contactos en pulsos:



En el esquema anterior se han representado, además de los cables de los contactos del anemómetro y de la alimentación del circuito electrónico (Vcc y GND), los pines 4 y 5 de un adaptador de Entrada/Salida que permite a la NDS leer y escribir el estado de 32 pines, conectados a un registro de E/S de 32 bits, al cual se accede a través de la dirección absoluta de memoria `0x0A000000`. El pin 5 permite leer el estado interno del J-K (salida Q), mientras que el pin 4 permite realizar un reset del J-K (\sim RST, señal negada).

Según el fabricante del anemómetro, la forma del cabezal hace que su frecuencia de rotación se incremente en 1 Hz por cada 2,5 mph (millas por hora) que acumule la velocidad del viento. Asumiendo que 2,5 mph equivalen a 1,1176 m/s (\approx 112 cm/s), los siguientes gráficos muestran diversos ejemplos de la forma de onda que obtendremos en la entrada CLK del biestable para 4 frecuencias de ejemplo, junto con sus velocidades del viento equivalentes:



Como se puede observar, el ancho del pulso invertido (1-0-1) se reduce a medida que la frecuencia aumenta (aprox. $\tau/10$). Suponiendo que la frecuencia máxima de captura será de 50 Hz (55,88 m/s \approx 200 Km/h), el tiempo de pulso podrá ser de hasta 2 ms (milisegundos) como mínimo. Por este motivo, el biestable J-K tiene la entrada J conectada a 1, la entrada K conectada a 0 y la señal del pulso conectado a la entrada CLK (flanco ascendente), lo cual permitirá memorizar un 1 cada vez que el pulso pase de 0 a 1. De este modo, la NDS dispondrá de suficiente tiempo (τ mínimo = 20 ms) para detectar el 1 en el pin 5, y volver a poner el biestable a 0 con la señal de reset invertido, es decir, poniendo el pin 4 a 0. Además, la señal de reset se debe mantener a 0 durante 10 ms, y luego se debe volver a poner a 1 para permitir capturar el siguiente pulso.

Como la interfaz de Entrada/Salida utilizada no genera interrupciones, se deberá utilizar la RSI del *timer* 0, para la cual se propone utilizar una frecuencia de activación de 100 Hz.

El programa a implementar deberá ir realizando una serie de tareas independientes mientras efectúa la gestión del biestable J-K para medir la velocidad actual (instantánea) del viento. El valor de la velocidad se debe ir escribiendo por la pantalla inferior de la NDS, cada 3 segundos (aprox.), expresado en m/s con 2 decimales; cada medición debe ir precedida por un número correlativo, que empezará en 1 (ver pantalla del esquema).

Se dispone de las siguientes rutinas, ya implementadas:

Rutina	Descripción
swi 9	Rutina de la BIOS para división entera; parámetros (R0: dividendo, R1: divisor); resultado (R0: cociente, R1: resto, R3: cociente absoluto)
inicializaciones()	Inicializa el <i>hardware</i> (pantalla, etc.)

<code>tareas_independientes()</code>	Tareas que no dependen del anemómetro (ej. captación de otros parámetros meteorológicos, como temperatura, humedad relativa, presión atmosférica, etc.); tiempo de ejecución < 1 s
<code>swiWaitForVBlank()</code>	Espera hasta el próximo retroceso vertical
<code>printf(char *format, ...)</code>	Escribe un mensaje en la pantalla inferior de la NDS

Para programar la frecuencia de activación de la RSI del *timer* 0, se pide realizar la siguiente rutina específica:

```
void inicializar_timer0_01(unsigned int frec);
```

la cual recibe por parámetro la frecuencia de salida del *timer* 0. Esta rutina debe calcular el divisor de frecuencia asociado a la frecuencia de salida requerida, utilizando la segunda frecuencia de entrada más alta disponible (523.656,96875 Hz). Recordar que las direcciones de memoria de los registros de datos y de control del *timer* 0 se definen simbólicamente como `TIMER0_DATA` y `TIMER0_CR`, respectivamente, y que en el registro de control hay que activar los bits 7 y 6 para iniciar el *timer* y generar interrupciones, mientras que en los bits 1 y 0 hay que indicar el código de la frecuencia de entrada requerida, que en nuestro caso será 01 (de aquí el sufijo en el nombre de la rutina).

Se pide:

Programa principal y variables globales en C, RSI del *timer* 0 y rutina `inicializar_timer0_01()` en ensamblador.