

## Problema 2: Detector de inclinación

Se propone trabajar con el sensor de movimiento MK6, que se puede conectar a la NDS como tarjeta de expansión GBA. El programa a realizar deberá calibrar el sensor (solo al inicio de su ejecución) y después deberá representar gráficamente la inclinación de la NDS, ininterrumpidamente.



Se dispone de las siguientes rutinas, ya implementadas:

<i>Rutina</i>	<i>Descripción</i>
<code>inicializaciones()</code>	Realiza inicializaciones del <i>hardware</i>
<code>tareas_independientes()</code>	Tareas que no dependen del cálculo de la inclinación (ej. medir una distancia con un sensor láser)
<code>swiWaitForVBlank()</code>	Espera retroceso vertical
<code>dibujar_inclinacion(short inc_x, short inc_y)</code>	Dibuja en pantalla una burbuja en un nivel circular según los grados de inclinación que se pasan por parámetro
<code>calibrar_inclinacion(short *calib)</code>	Devuelve por referencia los valores para la calibración del cálculo de la inclinación

El contenido de un vector de calibración se estructura en las siguientes posiciones de 16 bits cada una:

<i>Posición</i>	<i>Campo</i>	<i>Descripción</i>
0	<code>xoff</code>	Offset de la aceleración X
1	<code>yoff</code>	Offset de la aceleración Y
2	<code>zoff</code>	Offset de la aceleración Z

3	<code>xsens</code>	Sensibilidad de la aceleración X
4	<code>ysens</code>	Sensibilidad de la aceleración Y
5	<code>zsens</code>	Sensibilidad de la aceleración Z

Además, hay que realizar la captura de la inclinación por interrupciones. Como el sensor de movimiento no genera interrupciones, habrá que utilizar las interrupciones del *timer 0*.

Se dispone de una rutina ya implementada de nombre `inicializar_timer0()` que programa la interrupción `IRQ_TIMER0` a una frecuencia aproximada de 10 Hz.

También disponemos de las siguientes rutinas para comunicarnos con el sensor de movimiento:

<i>Rutina</i>	<i>Descripción</i>
<code>iniciar_MK6()</code>	Activa el <i>chip select</i> del sensor de movimiento
<code>enviar_MK6(byte comando)</code>	Envía un comando al sensor de movimiento; concretamente, tendremos que enviar 2 comandos: Read_X: 0x00 Read_Y: 0x02
<code>short recibir_MK6()</code>	Recibe un short (16 bits) del sensor de movimiento, con el valor de aceleración del eje indicado con el comando anterior
<code>finalizar_MK6()</code>	Desactiva el <i>chip select</i> del sensor de movimiento

El protocolo de comunicación es el siguiente:

- `iniciar MK6`
- `enviar comando Read_X`
- `recibir, transformar y almacenar inclinación X`
- `enviar comando Read_Y`
- `recibir, transformar y almacenar inclinación Y`
- `finalizar MK6`

El total de tiempo para realizar esta comunicación no supera los 100 microsegundos, es decir, se puede incluir dentro de una RSI. Por lo tanto, el protocolo de comunicación se realizará dentro de la RSI del *timer 0*, que guardará los valores de inclinación dentro de dos variables globales `inclin_X` e `inclin_Y`.

Además, hay que transformar el valor de aceleración de cada eje al valor de inclinación, con la siguiente fórmula (ejemplo para el eje X):

$$\text{incl\_X} = \text{incl\_X} + (\text{acel\_X} - \text{xoff}) * \text{xsens};$$

donde `incl_X` es el valor de inclinación (acumulada) y `acel_X` es el valor de aceleración (instantánea) que devuelve el sensor de movimiento, y `xoff`, `xsens` son los valores de calibración del eje X; la fórmula para el eje Y es análoga.

Se pide implementar una función específica para realizar este cálculo, de nombre `convertir_aceleracion(short *incl, short acel, short off, short sens)` donde `incl` (inclinación) se pasa por referencia y el resto de parámetros se pasan por valor.

### Se pide:

Programa principal en C, RSI del `timer0` y rutina `convertir_aceleracion()` en ensamblador.