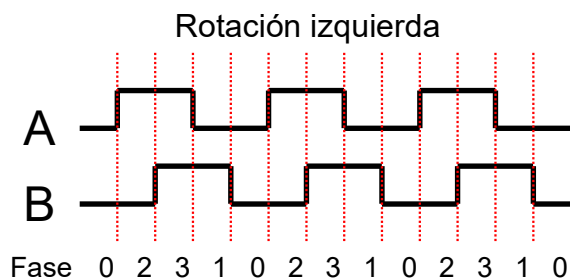
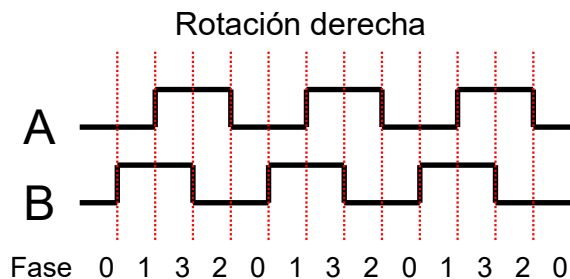
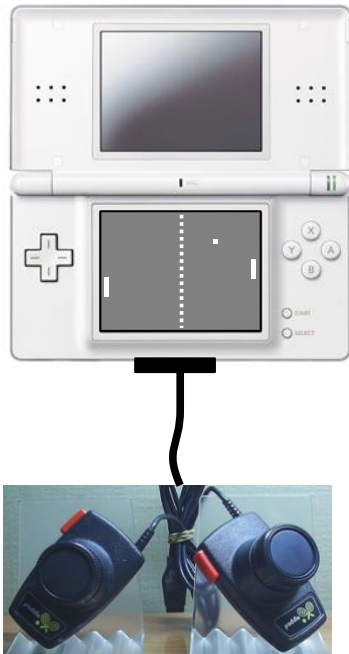


Problema 9: Paddles (Ex. 1ª Conv. 2011-12)

Se propone trabajar con un par de controles rotatorios de tipo *paddle*, que se puede conectar a la NDS como tarjeta de expansión GBA:



Estos controles permiten indicar velocidad y sentido de la rotación, usando simplemente 2 bits, de nombres A y B, que definen cuatro fases de la rotación, 0, 1, 2 y 3, según la codificación binaria del estado de los dos bits, donde A es el bit de más peso ($AB = 00 \rightarrow$ fase 0, $AB = 01 \rightarrow$ fase 1, etc.)

Cada vez que se cambia de fase indica el movimiento de una posición de la pala. Para detectar en qué sentido se mueve (incremento / decremento) hay que verificar si las fases siguen la secuencia (0, 1, 3, 2) o bien la secuencia inversa (0, 2, 3, 1).

El programa a realizar debe consultar periódicamente el registro de 16 bits específico que se encuentra en la posición $0 \times 0A000000$, el cual proporciona 3 bits por cada *paddle*:

Bit	Campo	Descripción
0	PADDLE1_C	Botón del <i>paddle</i> 1
1	PADDLE1_B	Bit B del <i>paddle</i> 1
2	PADDLE1_A	Bit A del <i>paddle</i> 1
3	PADDLE2_C	Botón del <i>paddle</i> 2
4	PADDLE2_B	Bit B del <i>paddle</i> 2
5	PADDLE2_A	Bit A del <i>paddle</i> 2

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 movimiento de los <i>paddles</i> (ej. controlar el movimiento de la pelota y el juego en general)
<code>swiWaitForVBlank()</code>	Espera retroceso vertical
<code>dibujar_raqueta(short posX, short posY)</code>	Dibuja una raqueta de juego según la posición indicada.

El programa principal consistirá en un juego de tenis: a parte de invocar a las tareas independientes, se encargará de dibujar dos raquetas con la posición X fijada por dos constantes (`posX1`, `posX2`) y la posición Y definida por dos variables globales (`posY1`, `posY2`).

Para realizar la detección del movimiento de los *paddles* se propone utilizar las interrupciones del *timer* 0, ya que el controlador de los *paddles* no genera interrupciones. Se dispone de una rutina ya implementada, de nombre `inicializar_timer0()`, que programa la interrupción `IRQ_TIMER0` a una frecuencia aproximada de 300 Hz.

La RSI del *timer* 0 tiene que detectar si hay un cambio en la fase de cada *paddle*. En caso afirmativo tiene que llamar a una rutina de nombre `detectar_sentido()`, que recibirá dos parámetros, la fase anterior y la fase actual. A partir de los dos valores de fase, la rutina devolverá 1 si el sentido es hacia la derecha, -1 si el sentido es hacia la izquierda, o 0 si los valores de fase no se corresponden a ninguna secuencia.

```
short detectar_sentido(char f_ant, char f_act);
```

Según el resultado de esta rutina, la RSI del *timer* 0 incrementará, decrementará o no hará nada sobre la variable global `posY` correspondiente.

Para detectar el sentido se recomienda utilizar dos vectores de bytes que indiquen, para cada número de fase anterior (índice del vector) cual es la fase actual siguiente para cada uno de los sentidos (contenido del vector):

```
char s_derecha[] = {1, 3, 0, 2};
char s_izquierda[] = {2, 0, 3, 1};
```

Se pide:

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