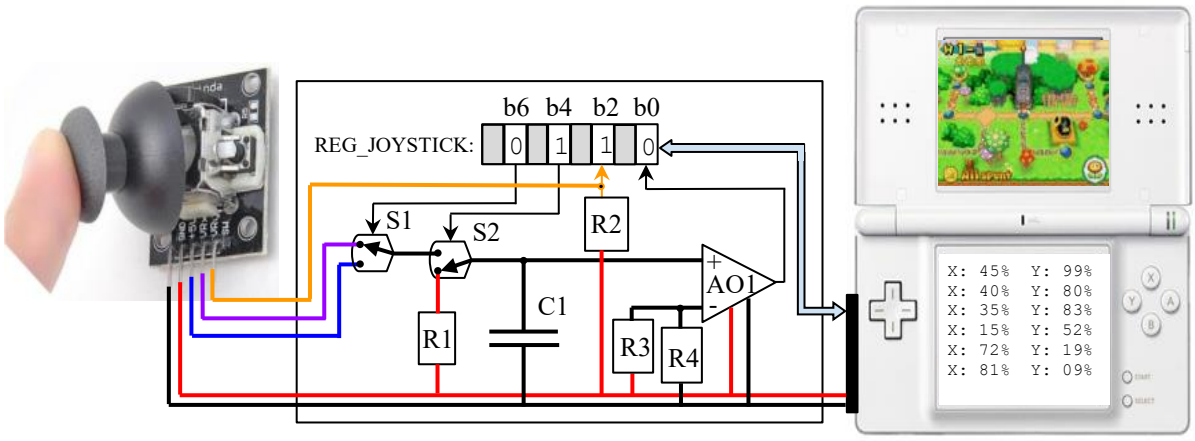
Problema 31: Joystick analógico (Ex. 1ª Conv. 2019-20)

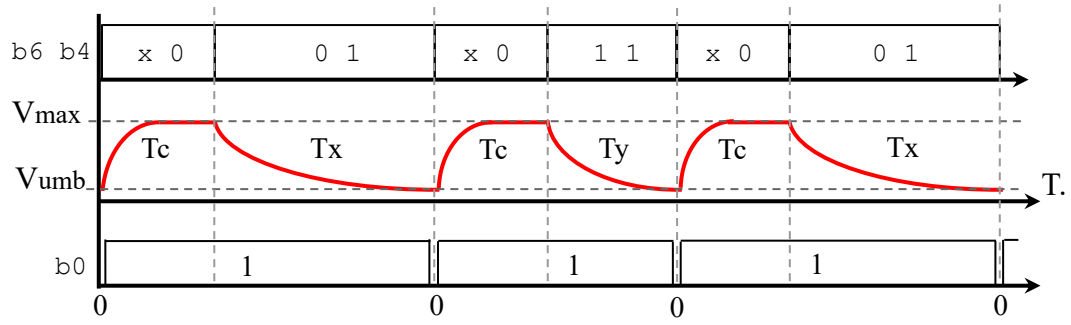
Se propone conectar a la NDS un *thumb joystick* (por ejemplo, sparkfun.com/9032). Este dispositivo consta de dos resistencias variables (potenciómetros) que se pueden regular con la inclinación del *stick*, una para cada eje de movimiento (X e Y). Además, dispone de un microinterruptor que se activa pulsando sobre el *stick*. Para medir el valor de las resistencias, se dispone de la siguiente interfaz electrónica:



La medición del valor de cada potenciómetro se basará en el tiempo de descarga del condensador C1 a través de dicho potenciómetro. La interfaz proporciona un único registro de Entrada/Salida de 8 bits, REG_JOYSTICK, que permite la gestión del circuito mediante los siguientes bits:

- `b6`: controla el *switch* `S1`, el cual permite conectar el condensador a uno de los dos potenciómetros (= 0 → `pot.X`, = 1 → `pot.Y`),
- `b4`: controla el *switch* `S2`, el cual permite conectar el condensador a la resistencia `R1` para el ciclo de carga, o a uno de los potenciómetros (según `S1`) para el ciclo de descarga (= 0 → carga, = 1 → descarga),
- `b2`: indica el estado del pulsador (= 0 → pulsado, = 1 → no pulsado),
- `b0`: indica si el condensador está prácticamente descargado (= 0) o tiene cierto nivel de carga (= 1); esta comprobación la realiza el amplificador operacional `AO1` que compara el voltaje del condensador con un voltaje umbral (V_{umb}).

El siguiente cronograma muestra un ejemplo de operativa del condensador:



En la secuencia anterior, los bits `b6` y `b4` se han manipulado de modo que el voltaje del condensador ha variado según la gráfica intermedia y el bit `b0` ha marcado el final de los tiempos Tx, Ty. Esta secuencia se puede representar con los siguientes estados:

estado	*b6 b4*	*Tiempo*	*Descripción*
0	x 0	Tc	carga del condensador durante un tiempo fijo
1	0 1	Tx	descarga del condensador a través del potenciómetro X, durante un tiempo variable (hasta que `b0` = 0)
2	x 0	Tc	carga del condensador durante un tiempo fijo
3	1 1	Ty	descarga del condensador a través del potenciómetro Y, durante un tiempo variable (hasta que `b0` = 0)
0	x 0	Tc	carga del condensador durante un tiempo fijo
1	0 1	Tx	descarga del condensador a través del potenciómetro X, durante un (nuevo) tiempo variable (hasta que `b0` = 0)
etc.			

El tiempo de carga Tc se fijará en 10 milisegundos, para asegurar que el voltaje del condensador llega a su valor máximo (V_{max}). La carga y descarga de un condensador sigue una ley exponencial que depende de los valores de las resistencias y la capacidad del condensador. Según la variación de resistencia de los potenciómetros, los tiempos de descarga Tx, Ty oscilarán entre 20 y 120 milisegundos (aprox.).

Se pide utilizar la RSI del *timer* 0 para fijar el valor de los bits `b6` y `b4`, así como para detectar el momento en que el bit `b0` pasa de 1 a 0, con el fin de obtener los tiempos de descarga. Para que dichos tiempos tengan una precisión suficientemente alta, el *timer* 0 estará programado a una frecuencia de 500 Hz.

Se dispone de las siguientes rutinas ya implementadas:

Rutina	*Descripción*
`inicializaciones()`	Inicializa el *hardware* (pantalla, interrupciones, timers, etc.)

`tareas_principales(`
`  unsigned char por_x,`
`  unsigned char por_y,`
`  unsigned char puls)`
	Tareas principales del programa (gestión de un juego); recibe por parámetro los porcentajes de las posiciones `x` e `y` del joystick, además de un booleano indicando si el botón está pulsado (tiempo de ejecución < 100 ms)

`swiWaitForVBlank()`	Espera hasta el próximo retroceso vertical
`printf(char *format,...)`	Escribe un mensaje en la pantalla inferior de la NDS

El programa a implementar consistirá en un juego, el cual se gestionará mediante llamadas consecutivas a la rutina `tareas_principales()`. Concurrentemente, la RSI deberá ir generando los estados de carga del condensador y descarga a través de los potenciómetros X e Y, alternativamente, además de capturar el estado del pulsador.

Después de cada ciclo completo de obtención de Tx y Ty, dichos tiempos se deberán convertir en un valor de porcentaje de variación del joystick en cada eje, donde 0% corresponderá a la posición mínima (izquierda en X, inferior en Y), 50% a la posición central y 100% a la posición máxima (derecha en X, superior en Y).

Para realizar dicha conversión será necesario disponer de los tiempos mínimos y máximos correspondientes a las posiciones mínimas y máximas de cada eje. Estos valores se almacenarán en las siguientes variables:

```
unsigned char tx_min, tx_max;          // tiempos límite (ms)
unsigned char ty_min, ty_max;
```

Para obtener estos valores con exactitud, ya que pueden variar respecto a los valores orientativos 20 y 120, el programa deberá realizar un proceso de calibración del joystick antes de que empiece el juego. Este proceso consiste en solicitar al usuario que posicione el joystick en la posición mínima y que pulse el botón del joystick. Cuando esto ocurra, habrá que memorizar los tiempos mínimos de cada eje. Este procedimiento se debe repetir para memorizar los tiempos máximos de cada eje.

La transformación continua de los tiempos variables Tx, Ty en valores de porcentaje se debe realizar con una regla de proporcionalidad entre el valor de tiempo mínimo y el valor de tiempo máximo (T = tmin → 0%, T = tmax → 100%). Esta transformación se debe realizar en el programa principal, para no sobrecargar innecesariamente el procesador dentro de la

ejecución de la RSI.

Por otro lado, cada vez que se realice una transformación de los dos tiempos variables, se deberán escribir los valores de porcentaje y un booleano que indique el estado del botón (0 → soltado, 1 → pulsado) por la pantalla inferior de la NDS, mientras que el juego se visualizará en la pantalla superior. A continuación se muestra una salida de ejemplo:

```
:
X: 45%   Y: 100%  P: 0
X: 40%   Y: 80%   P: 0
X: 35%   Y: 83%   P: 1
X: 15%   Y: 52%   P: 1
:
```

Se pide:

Programa principal y variables adicionales en C, RSI del *timer* 0 en ensamblador.