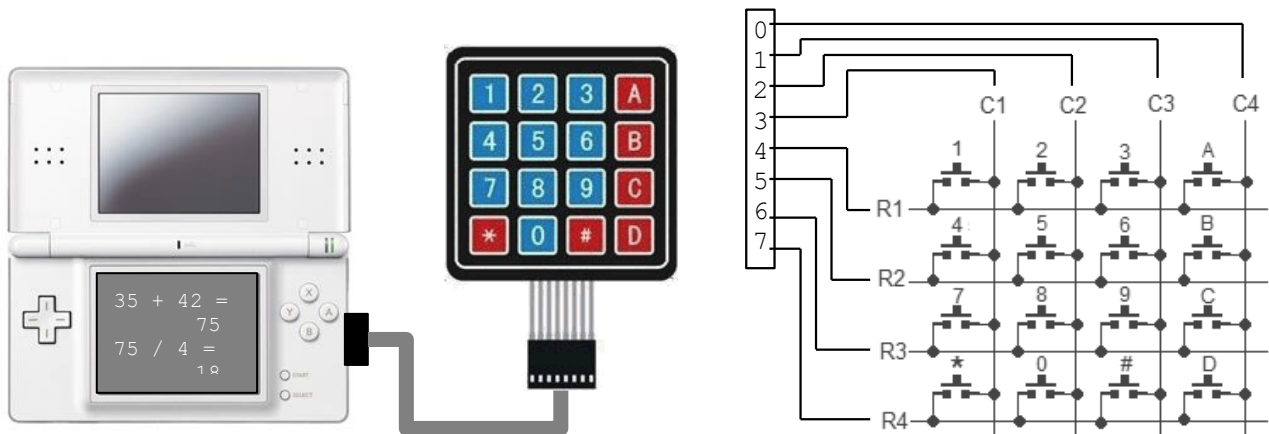


Problema 14: Teclado numérico (Ex. 1ª Conv. 2014-15)

Se propone controlar un teclado numérico de 16 teclas con la NDS. El teclado tiene una disposición matricial de los interruptores (teclas), en 4 filas por 4 columnas:



El dispositivo se conectará a la NDS mediante el puerto de cartuchos de juegos GBA ROM, y se controlará con un único registro de Entrada/Salida de 16 bits en la dirección simbólica REG_TECL, aunque solo los 8 bits de menos peso estarán conectados a la matriz de contactos, según indica el esquema de la figura anterior (bit 0 → C4, bit 1 → C3, etc.).

El registro es de lectura/escritura. La forma de leer las teclas será por barrido de filas y detección de columnas. Esto significa que, para cada fila, hay que realizar los siguientes pasos:

- escribir el registro REG_TECL, fijando todos los bits de filas (b7..b4) a 1 excepto el bit de la fila a analizar, que debe estar a 0,
- al cabo de cierto tiempo (del orden de centésimas de segundo), leer el registro REG_TECL, el cual presentará en cada bit de las columnas (b3..b0) un 0 si el interruptor correspondiente está pulsado, o un 1 si no lo está.

Después de analizar la última fila se debe volver a empezar por la primera, efectuando un barrido de todo el teclado a una frecuencia de 10 Hercios.

El programa de control debe consultar el estado de las teclas periódicamente y procesar cada pulsación para implementar una calculadora digital, escribiendo por pantalla los datos, operaciones y resultados generados.

Se dispone de las siguientes rutinas, ya implementadas:

<i>Rutina</i>	<i>Descripción</i>
<code>inicializaciones()</code>	Inicializa el <i>Hardware</i> (pantalla, interrupciones, <i>timer</i>)
<code>swiWaitForVBlank()</code>	Espera hasta el próximo retroceso vertical
<code>processKey(char key)</code>	Procesa la tecla que se le pasa por parámetro y realiza la función de calculadora, mostrando la información por pantalla; tiempo de ejecución < 0,01 s

Para poder capturar la pulsación de las teclas de forma concurrente con el programa principal, se pide utilizar la RSI del *timer* 0, que se programará (por la rutina de inicializaciones) para realizar 40 interrupciones por segundo.

En esta RSI hay que controlar el barrido de las 4 filas del teclado y guardar en una variable global, de nombre `currentKey`, un código numérico correspondiente a la tecla pulsada. Dicho código empezará por 0 para la tecla superior-izquierda (tecla “1”) y se incrementará de izquierda a derecha y de arriba a abajo del teclado. Si no hay ninguna tecla pulsada, la variable contendrá un -1. Si hay varias teclas pulsadas al mismo tiempo, sólo se almacenará el código de la tecla que tenga el número más grande (prioridad alta).

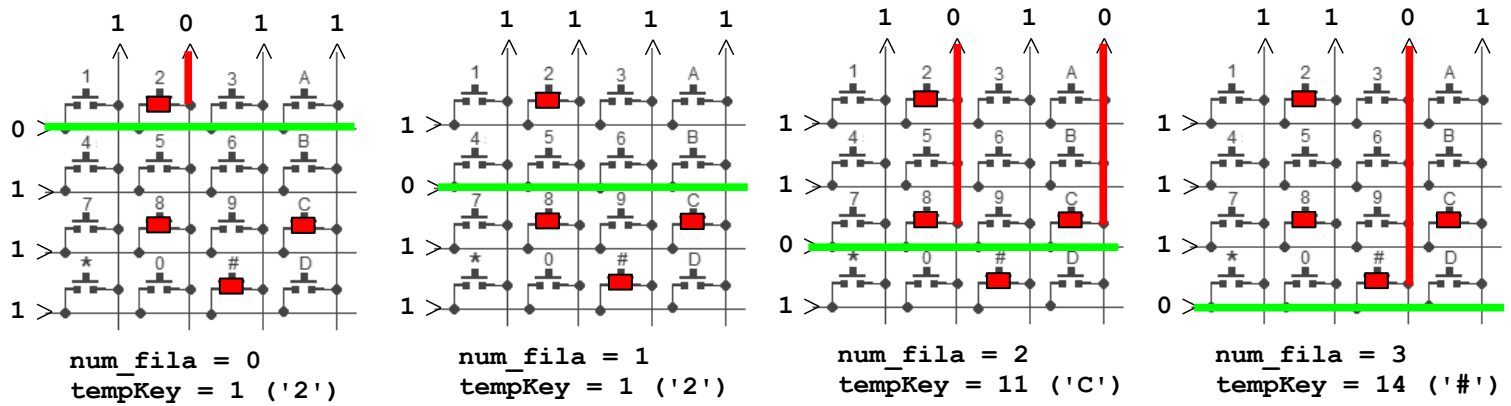
La RSI deberá llamar a una rutina auxiliar que se encargará de convertir los bits de columnas del registro `REG_TECL`, pasados por parámetro junto con el número de fila actual, en el código numérico de la tecla más prioritaria (código mayor) que se esté pulsando en dicha fila, o -1 si no existe pulsación en dicha fila:

```
char descodificar_tecla(char regteclval, char num_fila);
```

Por su parte, el programa principal debe sincronizarse con la RSI a través de la variable global `currentKey`, por el método de encuesta desde interrupción periódica, es decir, que se deben poder ejecutar las tareas independientes mientras se procesan las teclas. Además, para evitar que una misma pulsación se interprete como varias, será necesario detectar un cambio en `currentKey` antes de procesar el nuevo código de tecla con `processKey()`.

Por último, hay que tener en cuenta que la variable `currentKey` solo se actualizará al final del barrido de todo el teclado, puesto que hay que contrastar la prioridad de las teclas de las distintas filas, para lo cual necesitaremos otra variable global de nombre `tempKey` para mantener el código de tecla mayor de todo el barrido, además de la variable global `num_fila` que indicará la fila actual de procesamiento.

A continuación se muestra un ejemplo de barrido del teclado suponiendo que se han pulsado cuatro teclas a la vez ("2", "8", "C" y "#"). Aunque es un caso improbable, sirve para mostrar los valores que se obtendrán en cada fila y cómo se debe actualizar la variable `tempKey`:



Se pide:

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