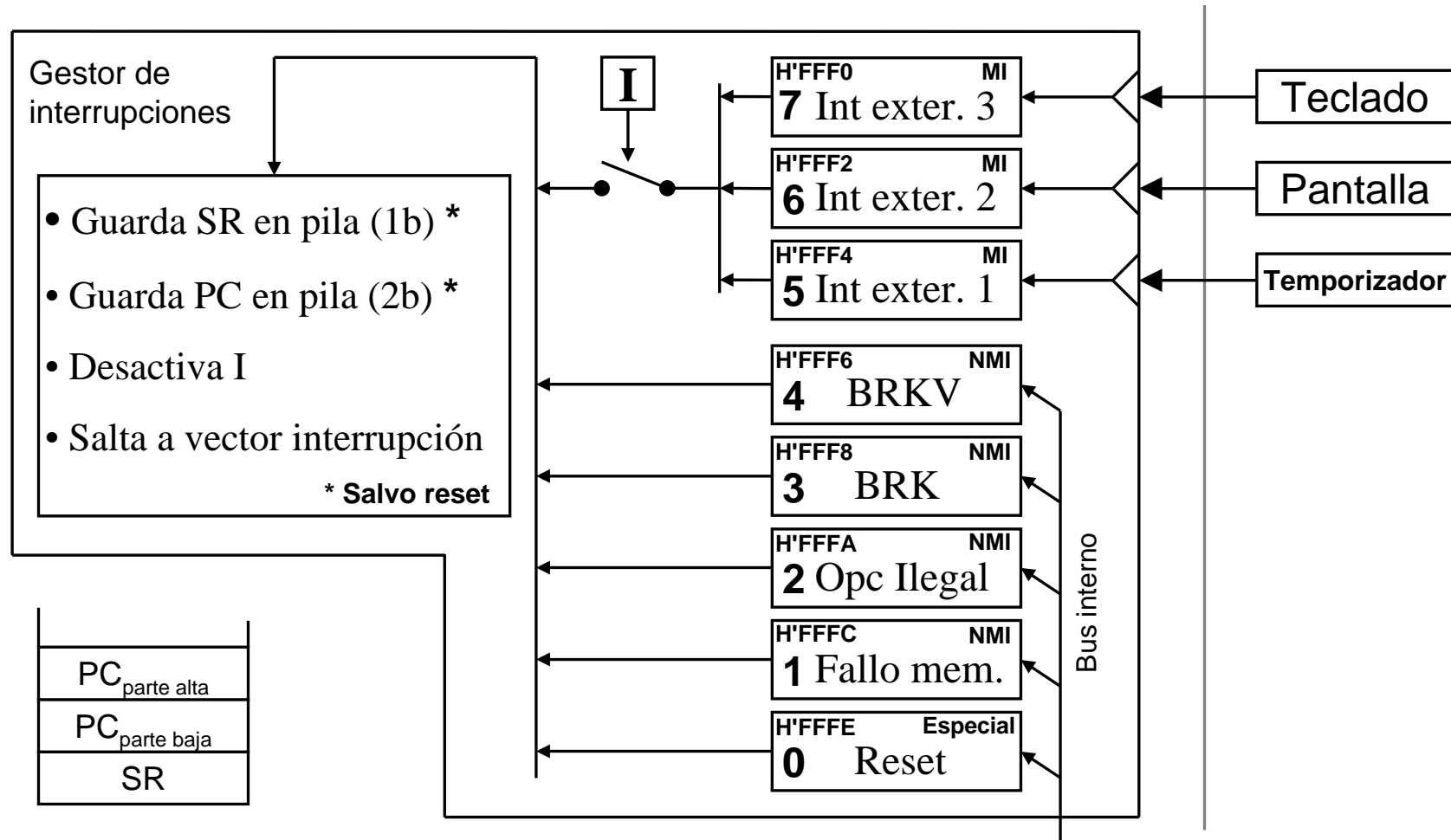


Programación en VCore - Parte 2

Interrupciones y manejo avanzado de periféricos

- Funcionamiento básico de las interrupciones en VCore.
- El teclado: un periférico simple.
- El temporizador y la medida de tiempo del sistema.
- La pantalla (tarjeta gráfica): un periférico complejo.
 - o Memoria de video y paginamiento.
 - o La paleta de colores.
 - o Generación de caracteres.

Funcionamiento básico de las interrupciones en VCore



Estructura de una rutina de servicio a interrupción

Nombre_Interrupcion:	
; Guardar registros que se van a modificar	
push	.0
push	.1
...	
Rutina de servicio al periferico que ha generado la interrupción	
; Recuperar registros guardados o. inverso	
...	
pop	.1
pop	.0
sti	; habilita interrupciones
rti	; retorna de interrupción

STI: Set I flag
(habilita interrupciones)
STI tiene un retardo de ejecución de una instrucción.

RTI: Return from Interrupt
(Retorna de la interrupción)

- Extrae de la pila la dirección de retorno (2 bytes) y la guarda en PC (similar a **rts**)
- Extrae de la pila el valor del registro de estado (1 byte) y lo guarda en SR.

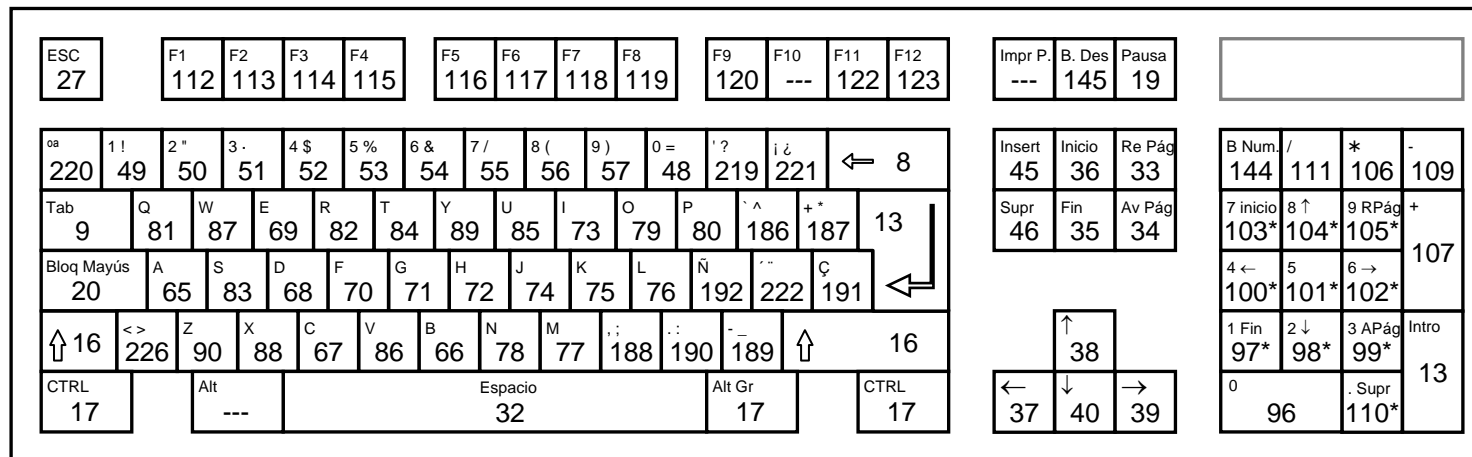
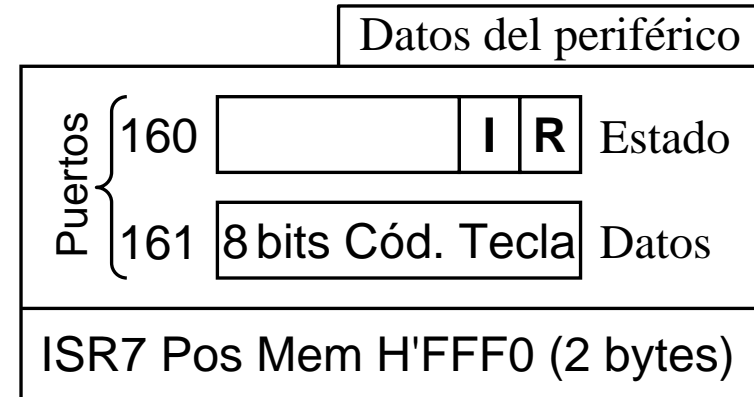
Nota importante: antes de habilitar interrupciones es necesario tener la pila preparada para recibir datos.

El teclado: un periférico simple

El teclado produce interrupción si:

- El usuario pulsa una tecla ($R == 1$)
- Interrupción del teclado activa ($I == 1$)
- Interrupciones CPU permitidas ($I_{CPU} == 1$)

Una lectura sobre el registro de estado del teclado libera la interrupción generada.



Cada tecla tiene un código asignado

* El código cambia en función del estado de Bloq Num

Gestión del teclado por cola de espera (buffer circular)

```

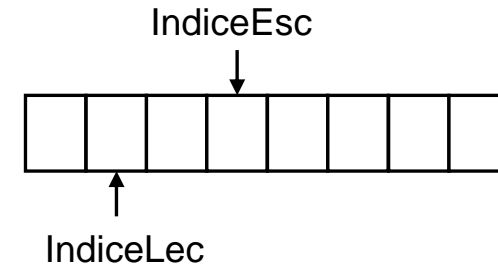
LongBufTec    equ    16
IndiceEsc:    dw     ? ;Indice de escritura
IndiceLec:    dw     ? ;Indice de lectura
BufTec1:      db     LongBufTec dup(?)

```

```

ISRTec:  push    .0    ;Guarda registros
          push    .1    ;a modificar
          ld      .0,/IndiceEsc
          ld      .1,.0  ; Calcula la
          inc     .0     ; siguiente pos
          cmp     .0,#LongBufTec1 ;del
          bnz     #CntISR ; puntero de
          clr     .0     ; escritura
CntISR:    cmp     .0,/IndiceLec
          bz      #FinISR
          st      .0,/IndiceEsc
          in      .0,/161 ;Lee tecla y
          st.b    .0,/BufTec1[.1] ;escr.
FinISR:    in      .0,/160 ;Libera intr.
          pop     .1     ;Recupera regs
          pop     .0     ;modificados
          sti     ;Activa ints.
          rti     ;Retorna d int

```



```

; Función: LeeTec
; Descripción: Lee tecla de buffer
; Parámetros: ninguno
; Devuelve: R0 tecla leida
LeeTec:  push    .1
          ld      .1,/IndiceLec
EspTec:    cmp     .1,/IndiceEsc
          bz      #EspTec
          ld.b    .0,/BufTec1[.1++]
          cmp     .1,#LongBufTec1
          bnz     #CntLee
          clr     .1
CntLee:    st      .1,/IndiceLec
          pop     .1
          rts

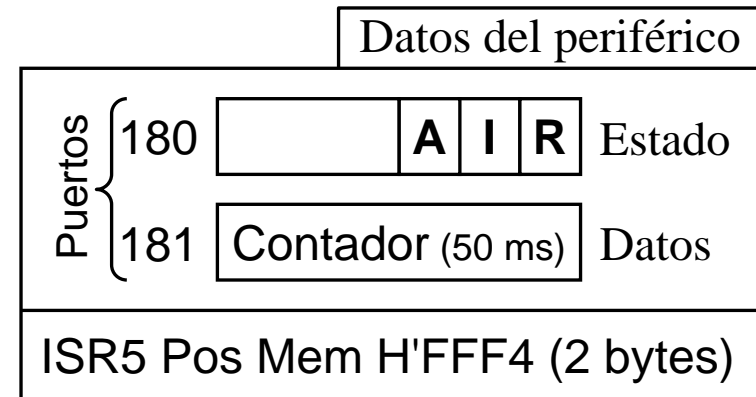
```

El temporizador: la medida de tiempo "real" del sistema

El temporizador produce interrupción si:

- El contador **llega** a 0 (Contador 1 \rightarrow 0)
- Interrup. temporizador activa ($I == 1$)
- Interrupciones CPU permitidas ($I_{CPU} == 1$)

Una lectura sobre el registro de estado del temporizador libera la interrupción generada.



Procedimiento para realizar un retardo de un determinado tiempo por espera activa:

- 1- Escribir 0 en el registro de estado para paralizar cualquier temporización en curso
- 2- Cargar el contador con el valor deseado de tiempo.
- 3- Activar el temporizador con $A = 1$ (no queremos interrupciones: $I = 0$)
- 4- Esperar a que R sea igual a 1.

Procedimiento para activar una interrupción temporizada:

- 1- Escribir 0 en el registro de estado para paralizar cualquier temporización en curso
- 2- Cargar el contador con el valor deseado de tiempo.
- 3- Activar el temporizador con $A = 1$ y la interrupción con $I = 1$.

Ejemplos de utilización del temporizador

```

;*****
; Función: EsperaTiempo (espera activa)
; Descripción: espera intervalo de tiempo
; Parámetros: R0 tiempo en unidades 50ms
; Devuelve: nada
EsperaTiempo:
    push    .1        ;Guarda R1
    clr     .1        ;Resetea
    out     .1,/180    ;temporiz
    out     .0,/181    ;Contador
    ld      .1,#4      ;Activa
    out     .1,/180    ;temporiz
    clr     .1        ;Espera a
BucleEspera:                ;temporiz
    in      .1,/180    ;terminada
    and     .1,#1      ;mirando R
    bz      #BucleEspera
    pop     .1        ;Recupera R1
    rts     ; Retorna a llamador

```

```

; Función: ISRTmp (interrupción)
; Descripción: interrupción cada segundo
ISRTmp:  push    .1        ;Guarda R1
         in      .1,/180    ;Baja int.
         ld      .1,#20    ;1 segundo
         out     .1,/181
         ld      .1,#6     ;Activa
         out     .1,/180    ;temporiz
         pop     .1        ;Recupera R1
         sti
         rti           ; Retorna de int

; Función: InitISR
; Descripción: inicia la interrupción
; Parametros: ninguno ; Devuelve: nada
InitISR: clr     .0
         out     .0,/180
         ld      .0,#20
         out     .0,/181
         ld      .0,#6
         out     .0,/180
         sti
         rts

```

La tarjeta gráfica: estructura general

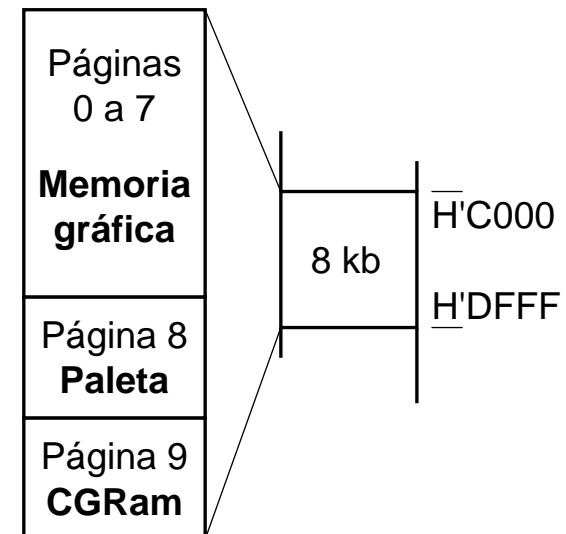
Puerto	Descripción	Formato					
192	Puerto de estado	<table border="1"><tr><td></td><td>V</td><td>S</td><td>I</td><td>R</td></tr></table>		V	S	I	R
	V	S	I	R			
193	Puerto de datos	<table border="1"><tr><td>Carácter ASCII</td></tr></table>	Carácter ASCII				
Carácter ASCII							
194	Selección de página	<table border="1"><tr><td>Página de video</td></tr></table>	Página de video				
Página de video							
195	Coordenada X del cursor	<table border="1"><tr><td>$0 \leq \text{Cur X} < 40$</td></tr></table>	$0 \leq \text{Cur X} < 40$				
$0 \leq \text{Cur X} < 40$							
196	Coordenada Y del cursor	<table border="1"><tr><td>$0 \leq \text{Cur Y} < 25$</td></tr></table>	$0 \leq \text{Cur Y} < 25$				
$0 \leq \text{Cur Y} < 25$							
197	Color escritura carácter	<table border="1"><tr><td>Color paleta</td></tr></table>	Color paleta				
Color paleta							
198	Color fondo carácter	<table border="1"><tr><td>Color paleta</td></tr></table>	Color paleta				
Color paleta							

Ind.	Acceso	Descripción
<div>R</div>	Lec	1: Pantalla lista para recibir carácter
<div>I</div>	Lec/Esc	1: Interrupciones pantalla permitidas
<div>S</div>	Esc	1: Fuerza reset en la tarjeta gráfica
<div>V</div>	Lec/Esc	1: Modo gráfico (fondo texto transpa.)

La tarjeta gráfica genera interrupción cuando se libera para la escritura de un nuevo carácter si I_{pantalla} e I_{CPU} están activadas.

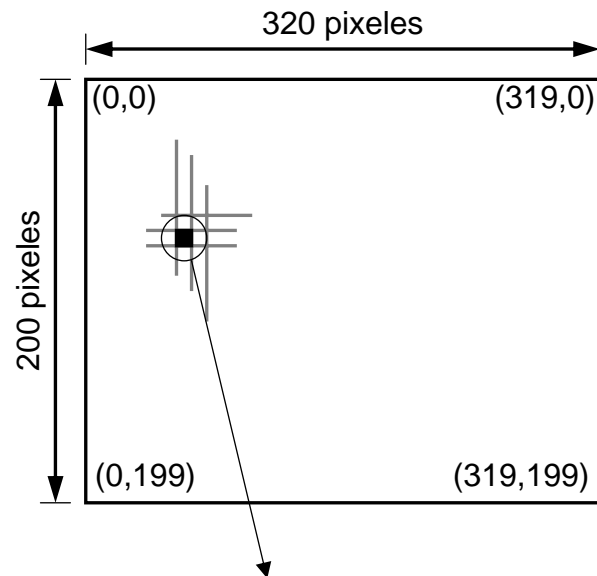
Una lectura sobre el puerto de estado libera la interrupción generada.

ISR6 Pos Mem H'FFF2 (2 bytes)



La memoria gráfica

Un pixel = Un byte de la memoria gráfica



$\text{Dir. memoria gráfica} = X + 320 \times Y$

$\text{Página} = \text{Dir. mem. gráfica} / 8192$

$\text{Dirección espacio VCore} = \text{H'C000} + \text{Dir. mem. gráfica} \bmod 8192$

```
; *****
; Función: PintaPixel
; Descripción: pinta punto en coordenadas (X,Y)
; Parámetros: R0 coord X; R1 coord Y; R2 color
; Devuelve: nada
PintaPixel:
    push .3          ; Guarda registros que
    push .1          ; van a ser modificados.
    ld .3,.1         ; Realiza multiplica. de
    asl .3,#8        ; coordenada Y por 320
    asl .1,#6        ; R = Y*320 = Y*(256+64)
    add .1,.3
    add .1,.0        ; y suma la coordenada X
    ld .3,.1         ; R3 = R1 = X+Y*320
    lsr .3,#13       ; Calcula número de pág.
    out .3,/194      ; para el pixel y ponlo
    and .1,#H'1FFF; Calcula la dirección
    add .1,#H'C000; de memoria VCore donde
    st.b .2,[.1]     ; se escribirá el pixel.
    pop .1           ; Recupera registros
    pop .3           ; previamente guardados
    rts              ; Retorna a llamador
```

La paleta de colores

Paleta de 256 colores configurable en página 8

n	Red	Grn	Blue	Nombre
0	H'00	H'00	H'00	Negro
1	H'00	H'00	H'80	Azul
2	H'00	H'80	H'00	Verde
3	H'00	H'80	H'80	Cián

n	Red	Grn	Blue	Nombre
4	H'80	H'00	H'00	Rojo
5	H'80	H'00	H'80	Magenta
6	H'80	H'80	H'00	Marrón
7	H'C0	H'C0	H'C0	Grís claro

Color 0	⁰ R	¹ G	² B
Color 1	³ R	⁴ G	⁵ B
Color 2	⁶ R	⁷ G	⁸ B
Color 3	⁹ R	¹⁰ ...	

Generación de caracteres

Juego de caracteres configurable en página 9.

Si la tarjeta está en modo gráfico, el color de fondo del carácter no se pintará.

