# Módulo 2. Entrada/salida

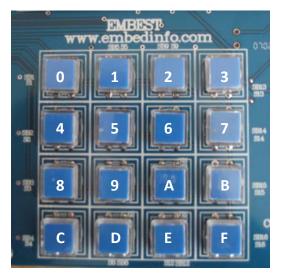
## Teclado Matricial

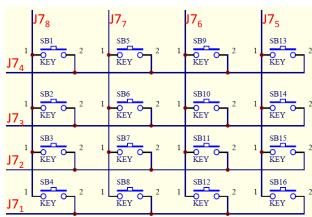
Guadalupe Miñana Ropero

# Teclado matricial: Disposición hardware



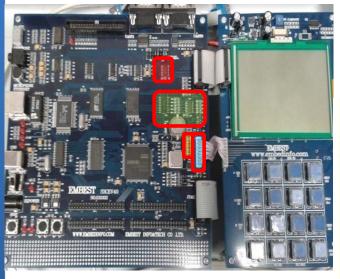
- Array bidimensional de pulsadores (SB1 –SB16)
- Cuando se pulsa una tecla se establece continuidad eléctrica entre fila (J7i, i=1,2,3,4) y columna (J7j, j=5,6,7,8)
- En el teclado nosotros suponemos el valor de las teclas como están en la figura

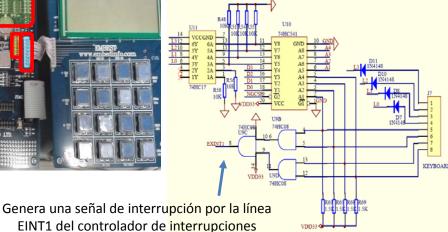




## Teclado matricial: Conexión con la placa y el S3C44B0X

- A diferencia de los controladores de E/S al uso, está lógica no dispone de registros de control, estado y configuración
- El teclado está conectado con la placa mediante un conector J7 y desde este conector a varios Cl





ec

# Teclado matricial: Conexión con la placa y el S3C44B0X

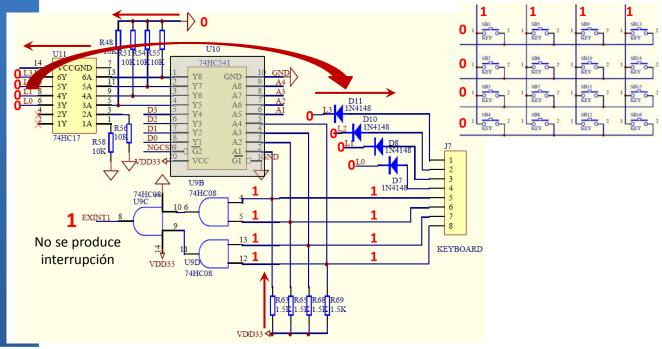


- Al controlador del teclado se accede por las dirección del banco 3 de memoria
  - Rango de direcciones: 0x06000000 0x07FFFFFF
  - Si en el bus de direcciones se pone una dirección del banco 3 la señal NGCS3 se pone a "0" y por lo tanto activa el módulo U10
- Genera una interrupción por la línea 1 del controlador de interrupciones (EINT1)
  - Recordad que las señales EINTO-EINT7 las maneja el puerto G del controlador
    GPIO

### **Teclado matricial: Estado inicial**



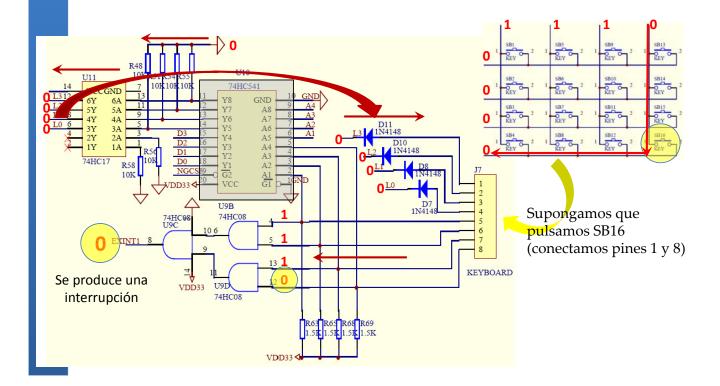
- En el bus de direcciones no hay una dirección del banco 3 de memoria
  - Módulo U10 desactivado
  - Módulo keyboard (J7)
    - pin 1= fila3, pin 2= fila2, pin 3= fila1, pin 4= fila0
    - pin 5= Columna3, pin 6= Columna2, pin 7= Columna1 y pin 8= Columna0



# Teclado matricial: Si se pulsa una tecla



En la situación anterior si se pulsa una tecla se produce una interrupción



## Teclado matricial: Si se pulsa una tecla



- Si se pulsa una tecla se genera una interrupción por EINT1
- Una vez detectada la interrupción comienza a ejecutarse la RTI
  - Esta subrutina tiene que identificar la tecla pulsada
  - Para averiguar qué tecla ha sido pulsada usamos la técnica de Scanning :
    - 1. Poner a "0" una fila y a "1" el resto de filas
    - 2. Leer el valor que tienen las columnas. Hay alguna columna a "0"?
      - Sí. Tecla identificada → la correspondiente a la fila y la columna con valor "0"
      - No → volver al paso 1

ec

# Técnica scanning



- Para poner un determinado valor en las filas del teclado hay que usar el módulo U10
  - Para activar el módulo U10 la dirección del bus tiene que pertenecer al rango del banco 3 de memoria (0x06000000 – 0x07FFFFFF)
    - Vamos a usar la primera dirección → 0x06000000
  - En los bits A4, A3, A2 y A1 (A0 vale 1 en este circuito) del bus de direcciones se pone el valor que queremos dar a las filas
  - Luego para poner:
    - Un cero en la fila 0: A4=1, A3=1, A2=1, A1= 0  $\rightarrow$  0x06000000 + 0xFD
    - Un cero en la fila 1: A4=1, A3=1, A2=0, A1= 1 → 0x060000000 + 0xFB
    - Un cero en la fila 2: A4=1, A3=0, A2=1, A1= 1 → 0x06000000 + 0xF7
    - Un cero en la fila 3: A4=0, A3=1, A2=1, A1=1→ 0x06000000 + 0xEF
- Para leer el valor que hay en las columnas hay que leer los bits del módulo U10 que están conectados el bus de datos (D0-D3)
  - Si D3=0, D2=1, D1=1 y D0= 1 (0x7) → Columna 0
  - Si D3=1, D2=0, D1=1 y D0= 1 (0xB) → Columna 1
  - Si D3=1, D2=1, D1=0 y D0= 1 (0xD) → Columna 2
  - Si D3=1, D2=1, D1=1 y D0= 0 (0xE) → Columna 3
  - Si D3=1, D2=1, D1=1 y D0= 1 (0xF) → No se ha pulsado ninguna tecla

# Técnica scanning

Valores del bus de	Valores del bus de datos						
dirección 0x06000000 +	0x7	0xB	0xD	0xE	0xF		
0xFD	SB1	SB5	SB9	SB13	-		
0xFB	SB2	SB6	SB10	SB14	-		
0xF7	SB3	SB7	SB11	SB15	-		
0xEF	SB4	SB8	SB12	SB16	_		

Tecla identificada

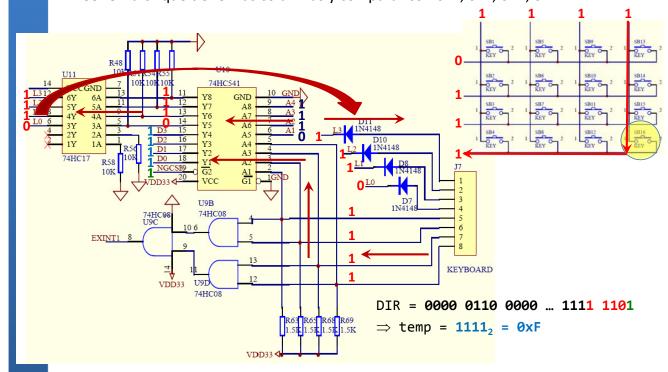


ec

# Técnica scanning: Detección de tecla (SB16)



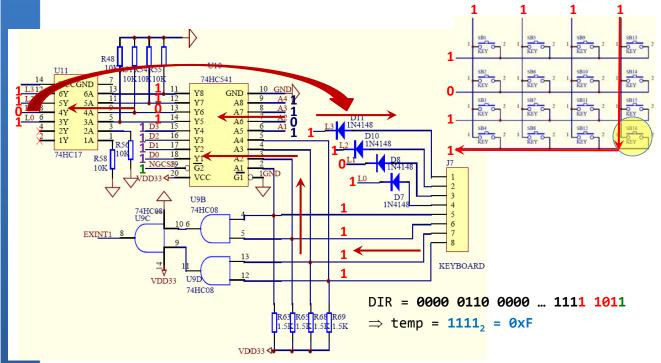
- Poner a "0" la fila cero y a "1" el resto  $\Rightarrow$  0x06000000 + 0xFD char temp = \*(0x06000000 + 0xFD)
- Leer el valor que tienen las columnas y comparar con 0x7, 0xB, 0xD, 0xE



# Técnica scanning: Detección de tecla (SB16)



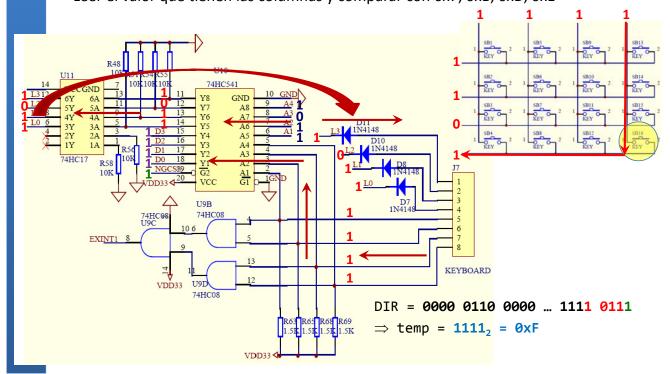
- Poner a "0" la fila cero y a "1" el resto  $\rightarrow$  0x06000000 + 0xFB
  - char temp = \*(0x06000000 + 0xFB)
- Leer el valor que tienen las columnas y comparar con 0x7, 0xB, 0xD, 0xE



# Técnica scanning: Detección de tecla (SB16)



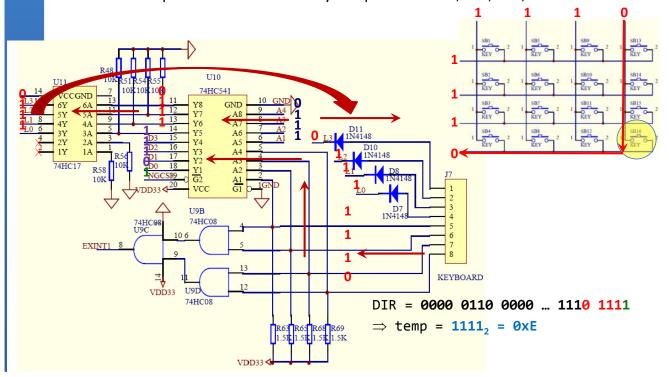
- Poner a "0" la fila cero y a "1" el resto  $\rightarrow$  0x06000000 + 0xF7 char temp = \*(0x06000000 + 0xF7)
- Leer el valor que tienen las columnas y comparar con 0x7, 0xB, 0xD, 0xE



# Técnica scanning: Detección de tecla (SB16)



- Poner a "0" la fila cero y a "1" el resto  $\rightarrow$  0x06000000 + 0xEF char temp = \*(0x06000000 + 0xEF)
- Leer el valor que tienen las columnas y comparar con 0x7, 0xB, 0xD, 0xE



# Rebotes al pulsar una tecla



- ¿Cómo evitar los rebotes que se producen al pulsar una tecla?
  - Rebotes de presión
    - Al entrar en la RIT que trata el teclado hay que esperar un tiempo antes de efectuar la identificación de tecla
  - Rebotes de depresión
    - Antes de salir de la RTI que trata el teclado
      - Esperar hasta que se detecte que se ha liberado la tecla pulsada
      - Esperar un tiempo



### void keyboard\_ISR (void)

- Función para tratar la interrupción del teclado, sigue el siguiente esquema:
  - Esperar 20ms para eliminar los rebotes de presión
  - Escanear el teclado utilizando la función int kb scan (void)
  - Poner en el display la tecla
  - Cambiar la configuración del timer 0 para que la generación de interrupciones periódicas tenga un periodo distinto en función de la tecla pulsada:
    - Tecla 0: periodo de 2s, valor de cuenta 62500 y divisor 1/8
    - Tecla 1: periodo de 1s, valor de cuenta 31250 y divisor 1/8
    - Tecla 2: periodo de 0.5s, valor de cuenta 15625 y divisor 1/8
    - Tecla 3: periodo de 0.25s, valor de cuenta 15625 y divisor ¼
    - Resto de teclas: no cambiaremos la configuración del timer
  - Esperar a que se deje de presionar la tecla leyendo el bit 1 del registro de datos del puerto G
    - No se puede usar la función portG\_read() porque sólo está implementada para los botones
    - Leer directamente el registro de datos del puerto G
  - Esperar 20ms para eliminar rebotes de depresión.
  - Borrar el flag de interrupción por la línea EINT1

### Conexión del teclado



#### Pulsadores

- Existen 2 botones (pulsadores)
- Accesibles a través del puerto G

#### Puerto G

- Registro de datos: PDATG
- Registro de configuración: PUPG
  - Permite activar o no una resistencia de pull-up1 por cada pin
- Registro de control: PCONG
  - Permite configurar los pines para
    - Ser de entrada
    - Ser de salida
    - Activar una petición de interrupción por las líneas EINT\* del controlador de interrupciones



poner 0 para activar la resistencia

--

Bit 14	Bit 13	12	Bit Bit 3 2

00 para configurar el pin como entrada 11 para activar interrupción EINT 6, 7 o 1 (línea 21) si se pulsa alguno de los botones (línea 24) si se pulsa alguna tecla

Registro	Dirección	R/W	Descripción	Valor por defecto
PCONG	0x01D20040	R/W	Configuración pines	0x0
PDATG	0x01D20044	R/W	Datos	Undef
PUPG	0x01D20048	R/W	Deshabilitar pull-up	0x0