

# TECNOLOGÍA ELECTRÓNICA DE COMPUTADORES

2º Curso – GRADO EN INGENIERÍA INFORMÁTICA  
EN TECNOLOGÍAS DE LA INFORMACIÓN

*Tema 10: Aplicaciones de los microcontroladores*

*Lección 23. Dispositivos de E/S: teclados y pulsadores*

## Lección 23. Dispositivos de E/S: teclados y pulsadores

### 23.1. Aspectos generales de pulsadores y teclados

### 23.2. Teclados

- Teclados lineales
- Teclados matriciales e identificación de las teclas pulsadas
- Ejemplo de conexión a un PIC y programación

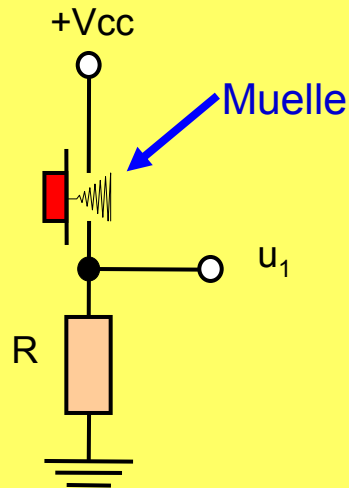
### 23.3. Otros dispositivos de entrada

- Microinterruptores
- Preselectores decimales o interruptores de décadas

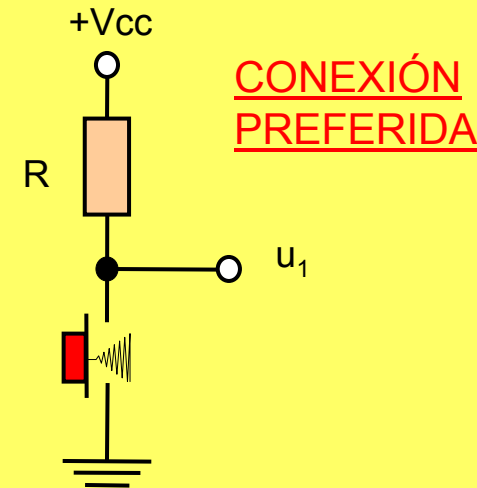
## 23.1. Aspectos generales de pulsadores y teclados

- Muy utilizados para introducir información al microcontrolador y otros circuitos digitales.
- Los hay de varios tipos: de lámina flexible, de efecto Hall, de efecto inductivo, de efecto capacitivo.
  - Los más comunes son los de lámina flexible.

### Conexiones típicas:



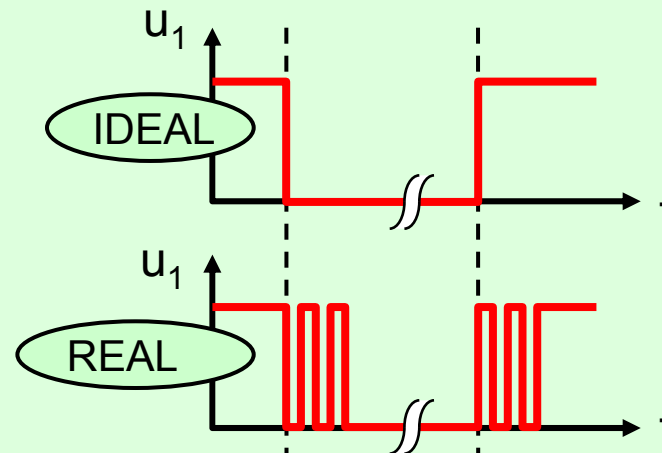
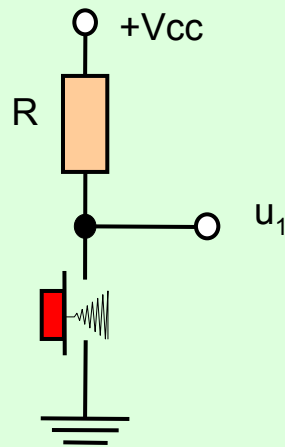
- Tecla pulsada = "1"
  - Tecla no pulsada = "0"
- En TTL el valor de R es crítico  
(ha de ser pequeño: máx 270  $\Omega$ )



- Tecla pulsada = "0"
- Tecla no pulsada = "1"

## Pulsadores y Teclados: el problema de los rebotes y soluciones

El problema de los rebotes: Debido al efecto muelle del pulsador, se producen oscilaciones en la señal tanto al pulsar como al soltar la tecla.

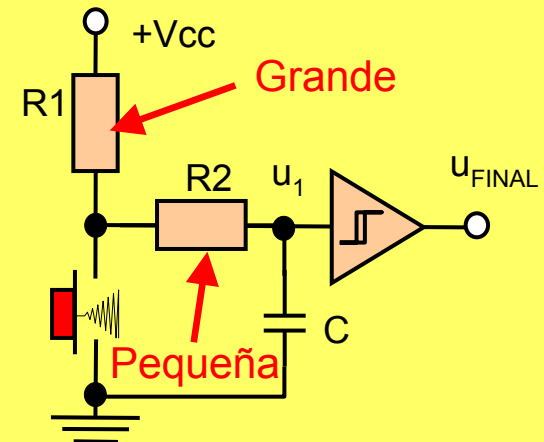
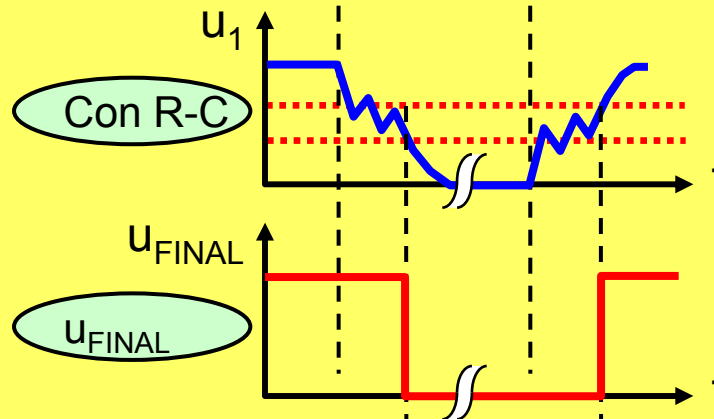


### SOLUCIONES

**Hard.:** Red R-C  
Biestables

**Soft:** Espera de un  
tiempo suficiente

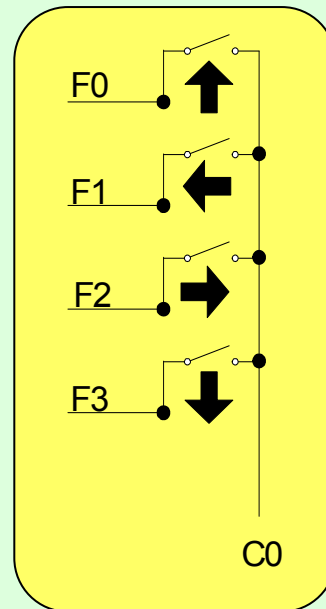
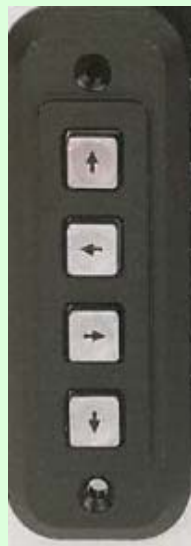
Solución con red  
RC y buffer  
Schmitt-Trigger.



## 23.2. Teclados

### Teclados Lineales

(Muy sencillos, pero no permiten disponer de muchas teclas)



#### Características técnicas típicas:

Valor nominal contactos: 24V d.c. y 50mA

Resistencia de contacto:  $< 500 \Omega$

Tensión de ruptura: 250V a.c.

Tiempo de rebote:  $< 10\text{ms}$ .

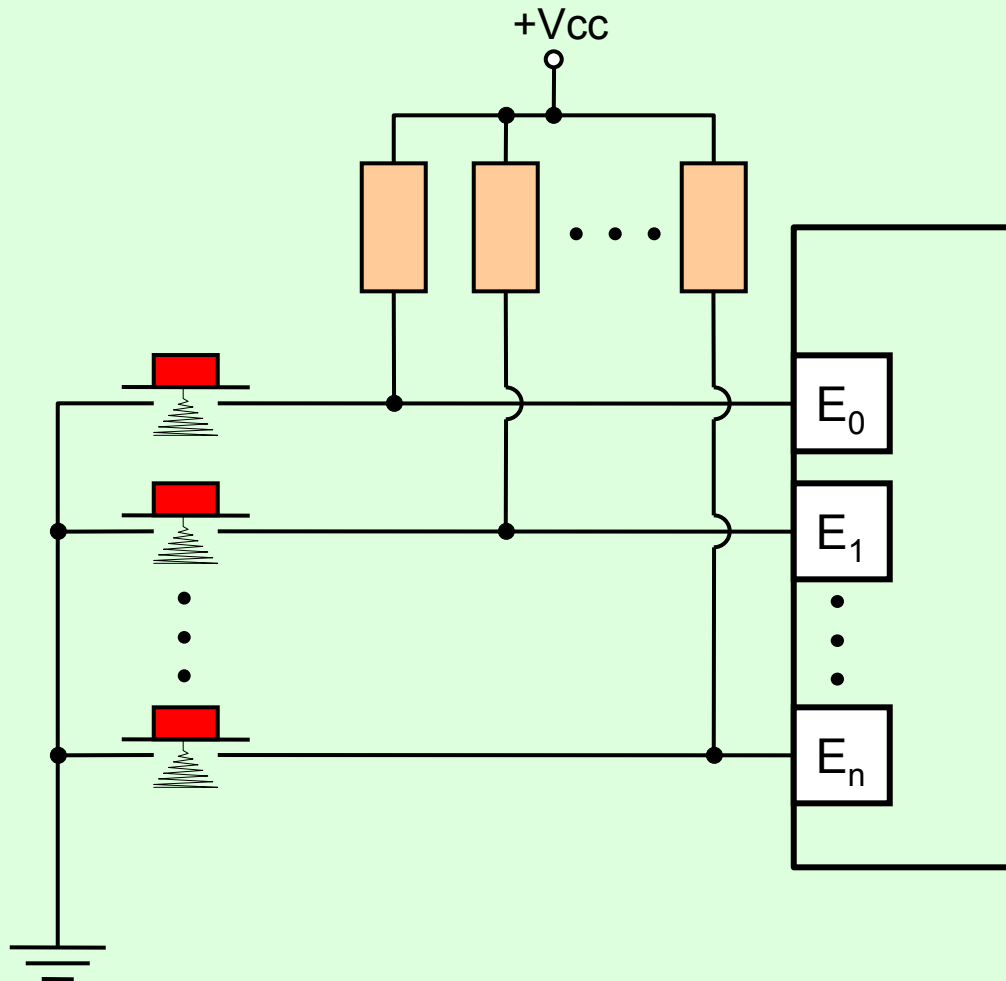
Vida útil mecánica:  $10^6$  operaciones

Temperatura

de funcionamiento:  $-20^{\circ}\text{C}$  a  $+55^{\circ}\text{C}$

de almacenamiento:  $-20^{\circ}\text{C}$  a  $+60^{\circ}\text{C}$

## Teclados lineales y conexión a un microcontrolador

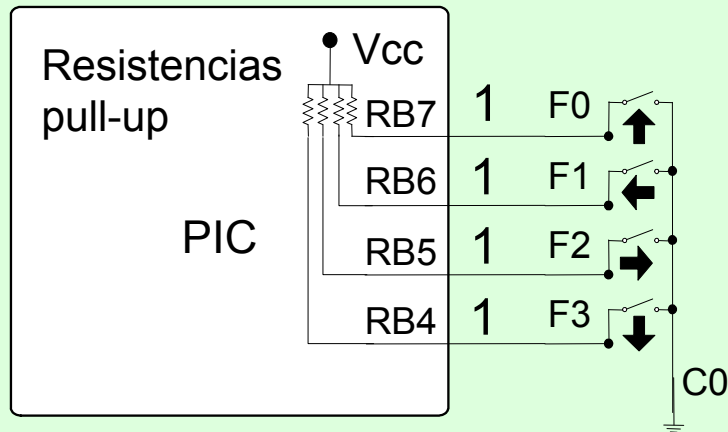


Basta con que el programa compruebe periódicamente el estado de las entradas a las que se ha conectado el teclado.

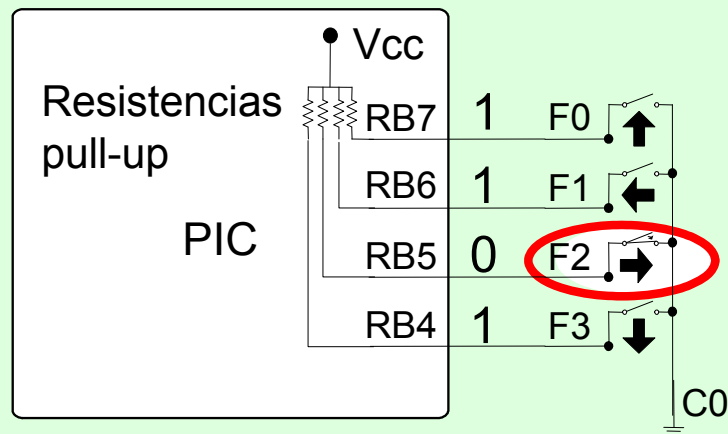


Cuando el microcontrolador detecte un "0" en una entrada, se sabrá la fila en la que se ha pulsado una tecla y, por tanto, se conocerá la tecla pulsada.

## Ejemplo: Conexión de un teclado lineal al PIC

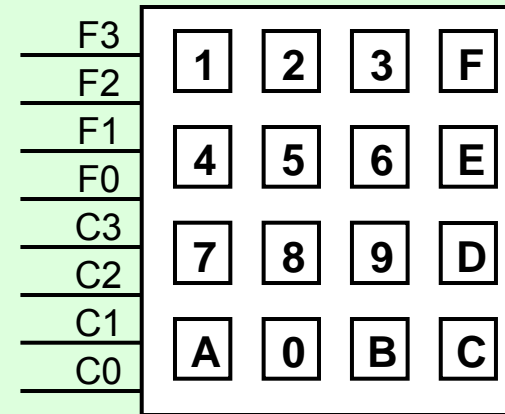


En un teclado lineal 4x1, la pulsación de una tecla provocará un cambio de valor en un bit del PORTB (RB5 en la figura).



Si RBIE y GIE están a “1”, este cambio provoca la entrada de la rutina de interrupción en la que se realizará la lectura del PORTB. El bit del nibble alto de PORTB que está a “0” identifica claramente la tecla pulsada.

## Teclados Matriciales



### Características técnicas típicas:

Valor nominal contactos: 24V d.c. y 50mA

Resistencia de contacto: < 500  $\Omega$

Tensión de ruptura: 250V a.c.

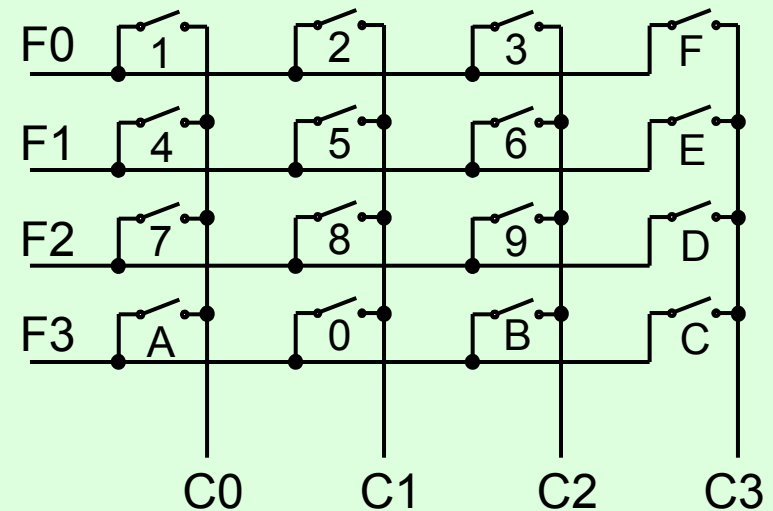
Tiempo de rebote: < 10ms.

Vida útil mecánica:  $10^6$  operaciones

Temperatura

de funcionamiento: -20°C a +55°C

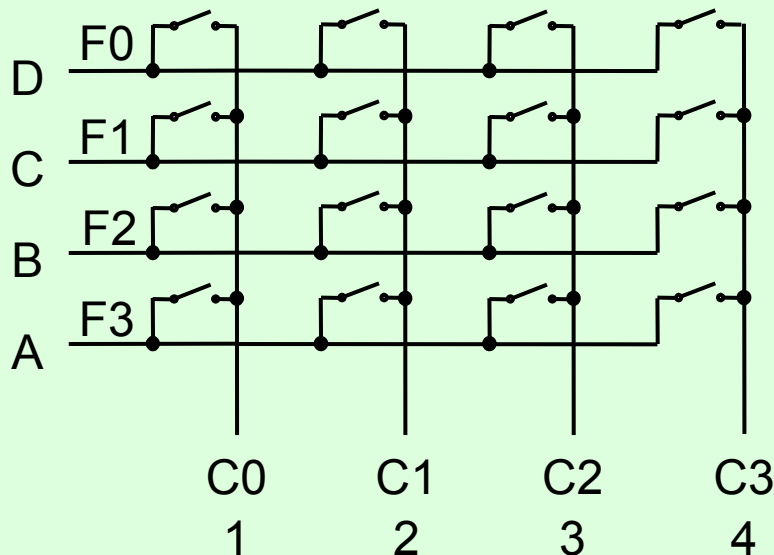
de almacenamiento: -20°C a +60°C



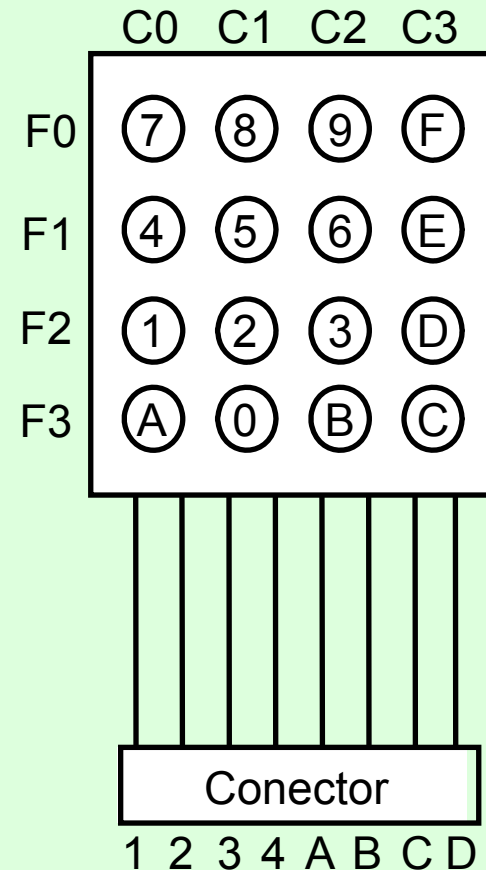


## Ejemplo: Teclado matricial del equipo de prácticas

Esquema Eléctrico

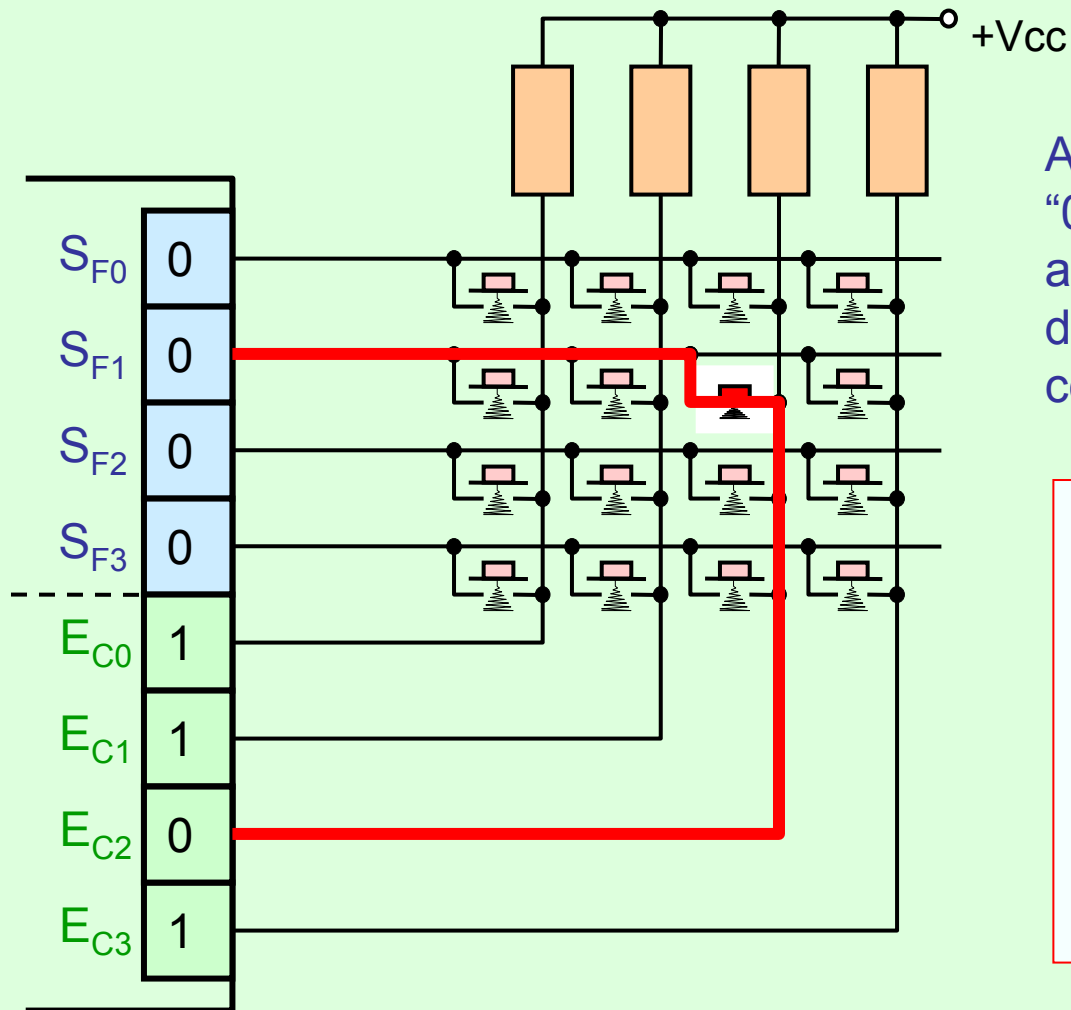


Disposición física de teclas



## Teclados matriciales

Ventaja: muchas teclas son controladas con un número reducido de líneas de E/S



Al estar todas las salidas a “0”, cuando se pulsa una tecla aparece un “0” en la entrada del micro conectada a la columna de la tecla pulsada.

En este ejemplo, se sabe que se ha pulsado una tecla de la tercera columna pero no se sabe cuál.

Se necesitan algoritmos que permitan determinar qué tecla se ha pulsado.

## Teclados matriciales: identificación de la tecla pulsada por software

### Fase 1: Detectar que se ha pulsado una tecla

- Interrupción
- Muestreo periódico (*polling*)

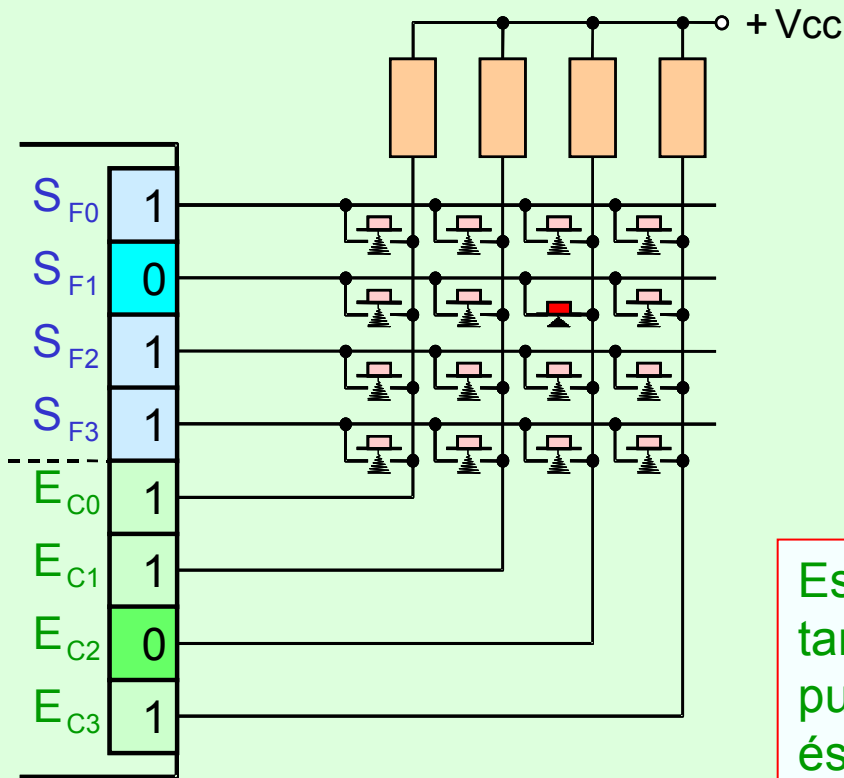
### Fase 2: Identificar qué tecla se ha pulsado

Técnicas de identificación de la tecla pulsada (fila y columna):

- Muestreo secuencial
- Inversión de línea

## Identificación por el Método del Muestreo secuencial

- Una vez que detectada la pulsación de una tecla, se va cambiando el valor de las salidas en el microcontrolador para que sólo una de ellas valga '0' en cada instante.
- La combinación que dé lugar a un '0' en alguna de las entradas identificará la tecla que se ha pulsado.



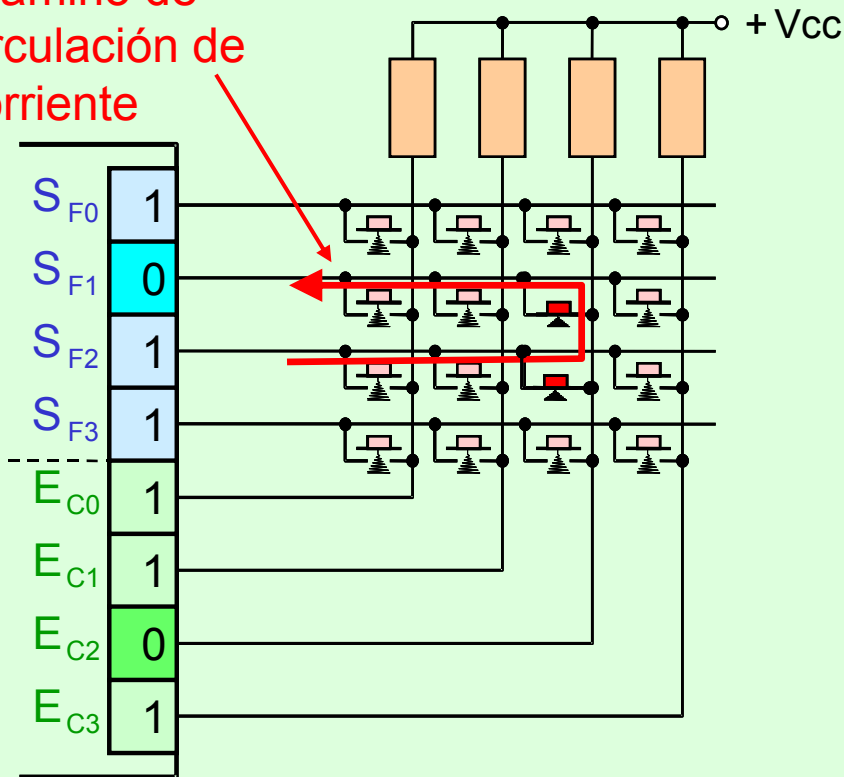
$S_{F0}$	$S_{F1}$	$S_{F2}$	$S_{F3}$	$E_{C0}$	$E_{C1}$	$E_{C2}$	$E_{C3}$
0	1	1	1	1	1	1	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1

Es un método sencillo de implementar, pero tardará más o menos en encontrar la tecla pulsada en función de la posición que ocupe ésta.

## Identificación por el Método del Muestreo secuencial

- Es recomendable proteger las salidas (mediante resistencias o diodos) ante la eventualidad de que se pulsen dos teclas de la misma columna a la vez
- Protección con resistencias

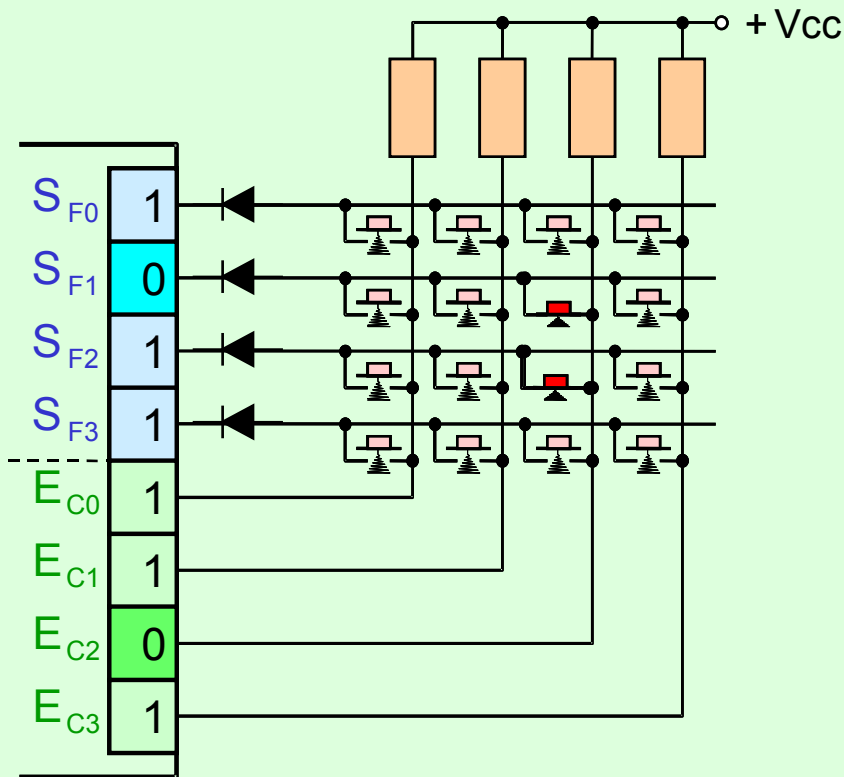
Camino de  
circulación de  
corriente



$S_{F0}$	$S_{F1}$	$S_{F2}$	$S_{F3}$	$E_{C0}$	$E_{C1}$	$E_{C2}$	$E_{C3}$
0	1	1	1	1	1	1	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1

## Identificación por el *Método del Muestreo secuencial*

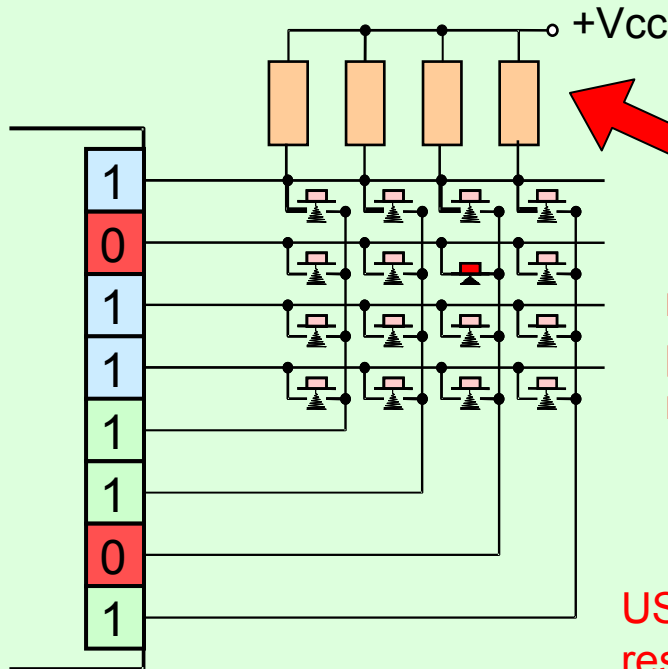
- Es recomendable proteger las salidas (mediante resistencias o diodos) ante la eventualidad de que se pulsen dos teclas de la misma columna a la vez
- Protección con diodos



$S_{F0}$	$S_{F1}$	$S_{F2}$	$S_{F3}$	$E_{C0}$	$E_{C1}$	$E_{C2}$	$E_{C3}$
0	1	1	1	1	1	1	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1

## Identificación por el Método de la Inversión de línea

- Tras detectar que hay una tecla pulsada, se almacena el valor que hay en las entradas, se invierten las líneas (las que eran entradas pasan a ser salidas y viceversa) y se saca por las nuevas salidas la combinación almacenada.
- Esto dará lugar a que en las nuevas entradas sólo aparezca un cero en la fila a la que pertenece la tecla pulsada.



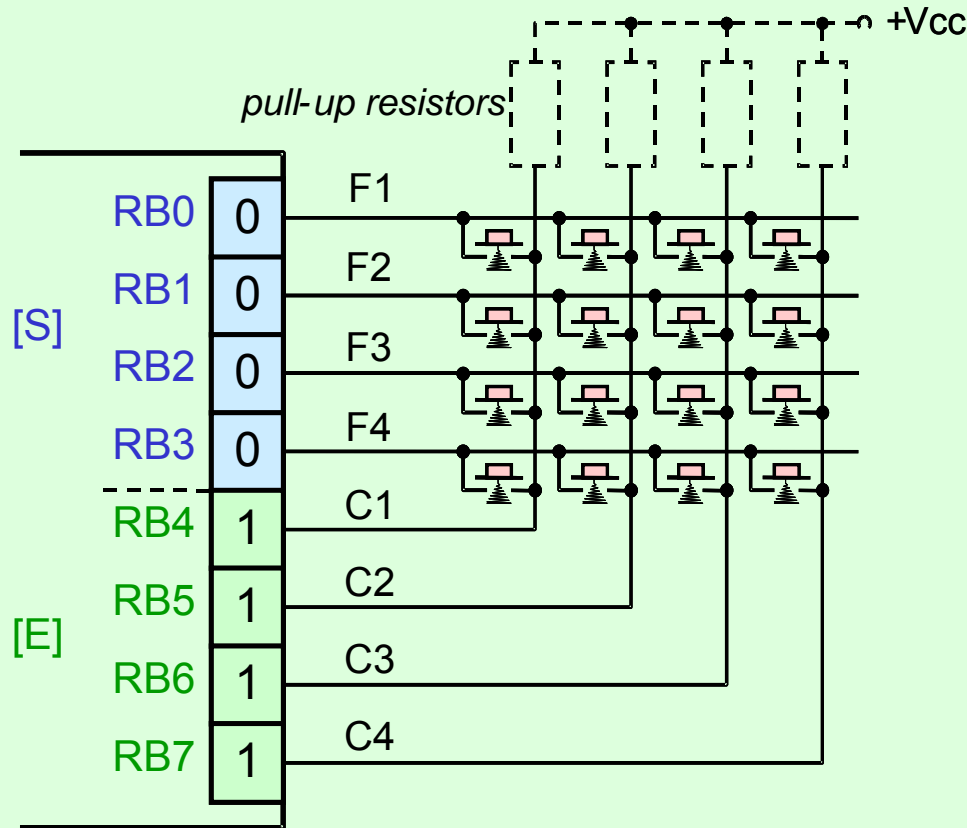
**!OJO!, deben existir resistencias de polarización en las nuevas entradas**

**USAR PORTB en los PIC:  
resistencias de *pull-up* en las  
entradas**

Este método es más rápido que el anterior y tarda lo mismo en identificar cualquier tecla.

## Conexión de teclados matriciales en los PIC

El puerto B de los microcontroladores PIC está especialmente pensado para conectar un teclado matricial de 4×4



### Comentario

- La posibilidad de habilitar resistencias de pull-up reduce el número de componentes externos.
- La existencia de una interrupción asociada a cambios en los bits RB<4:7> avisa de que se ha pulsado una tecla.



## Advertencia importante durante la depuración con MPLAB-ICD2

El Puerto B es el más indicado para conectar un teclado matricial 4×4 debido a la capacidad de habilitar resistencias de *pull-up* y de generar una interrupción cada vez que cambia el estado de alguno de sus cuatro bits más altos **RB<4:7>**.

Por ello se usa preferentemente este puerto para conectar estos teclados.

Sin embargo, debe tenerse en cuenta que el **debugger ICD2** hace uso de dos **pines del Puerto B** para llevar a cabo tareas de emulación (**RB7 y RB6**) y, por tanto, no se podrán manejar directamente con el microcontrolador

Esto se traduce en que, durante la emulación del sistema, algunas teclas “no funcionarán” según lo esperado: las filas o columnas conectadas a las líneas RB6 y RB7 del PORTB pueden tener niveles que dependen del ICD2 y no del teclado

Una vez grabado el programa definitivo en el microcontrolador, todas las líneas estarán operativas.

## Conexión de teclados matriciales en los microcontroladores PIC

### Ejemplo:

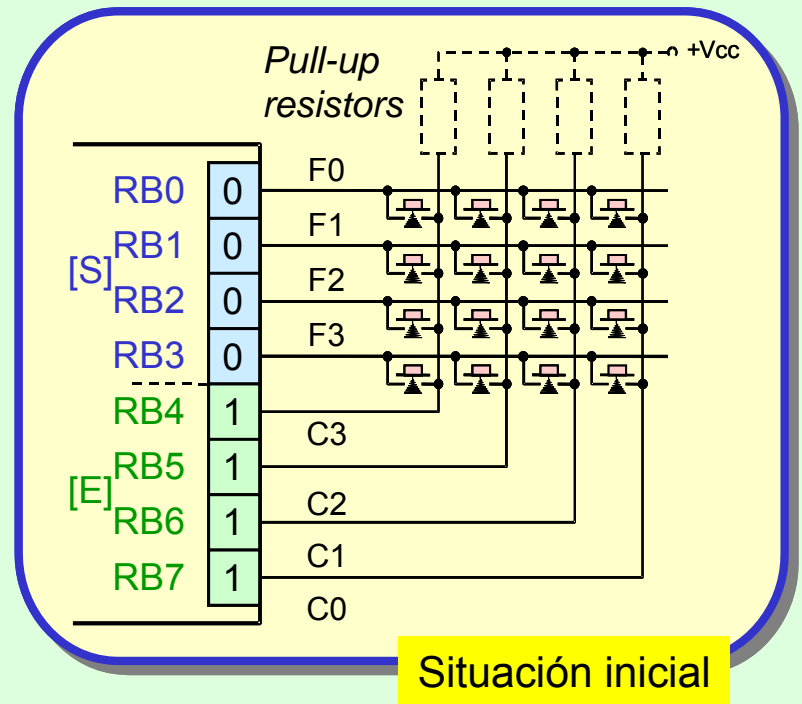
- Fragmento de Programa para PIC16F877 que utiliza un teclado matricial de 4 filas por 4 columnas (como el empleado en las prácticas) y que se conecta a la placa de entrenamiento PICDEM2-PLUS.
- El teclado se conecta al PORTB para detectar por interrupción la pulsación de alguna tecla.
- Se configuran las resistencias de polarización en las entradas para evitar conexionado adicional externo.
- Se identificará la tecla pulsada siguiendo el método de la inversión de línea.

## Activación de la interrupción para detección de la pulsación de una tecla

- Se activan las resistencias de pull-up de las entradas del PORTB.
  - Se definen las 4 líneas más altas del PORTB (RB7 a RB4) como entradas, en reposo se leerán como “1”.
  - Se definen las 4 líneas más bajas del PORTB (RB3 a RB0) como salidas y se ponen a “0”.
- Diagrama de configuración de los pines del PORTB:

Pin	Valor	Conexión
RB0	0	F0
RB1	0	F1
RB2	0	F2
RB3	0	F3
RB4	1	C3
RB5	1	C2
RB6	1	C1
RB7	1	C0

Se muestran también las resistencias de pull-up conectadas a +Vcc para las líneas de entrada (C1, C2, C3).
- Se pone a “0” el flag de detección de cambio en las líneas RB7 a RB4 del PORTB y se “leen” las líneas del PORTB para detectar cambios desde la última lectura.
  - Se activa la interrupción por cambio en cualquiera de las líneas RB4 a RB7.



## Ejemplo de entrada en el programa de tratamiento de interrupción

```

org 4                                ;Vector de interrupcion para las generadas
goto TECLA                          ;por cambio en RB4 a RB7

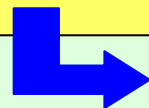
;*****
;
;Programa de tratamiento de la interrupción generada por cambio en PORTB,
;se producirá en cada cambio de las líneas RB4 a RB7
;*****
;

TECLA    btfss INTCON,RBIF            ;Si entramos aquí por otro motivo
        retfie                      ;distinto a RBIF=1 salimos de inmediato

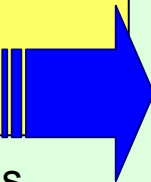
        movwf    W_tmp              ;Salvamos el registro W
        swapf    STATUS,W           ;el registro STATUS "girado" en W
        bcf      STATUS,RP0         ;Aseguramos el paso al banco 0
        bcf      STATUS,RP1
        movwf    STATUS_tmp         ;Guardamos en el banco 0
        movf     PCLATH,W           ;Salvamos también PCLATH
        movwf    PCLATH_tmp

;Espera para evitar rebotes en las teclas
call NO_REBOTE                      ;llamamos al subprograma de ESPERA

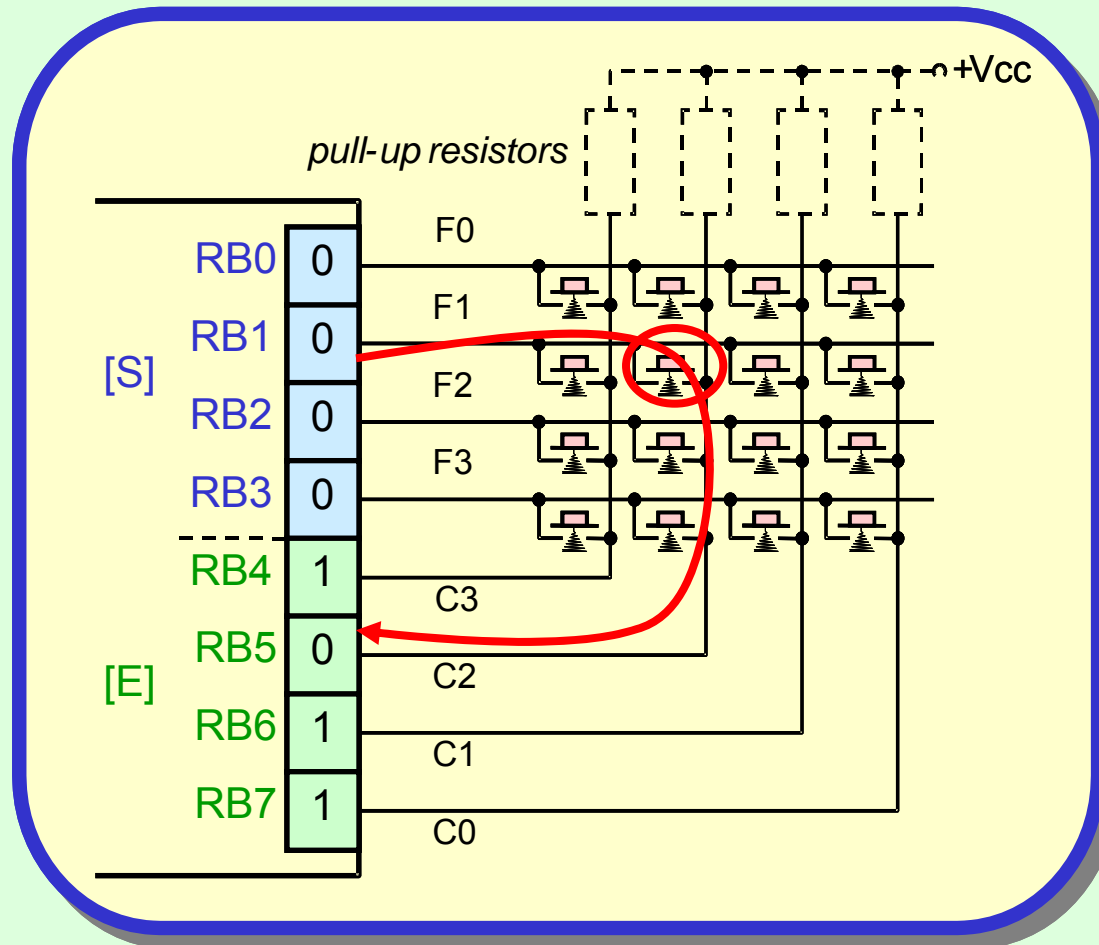
```



Subprograma de Espera para que desaparezcan rebotes



Ejemplo de entrada por pulsación de la tecla de la fila F1 y columna C2



La entrada en el programa de tratamiento se producirá tanto cuando se pulse como cuando se libere la tecla (hay cambio en ambos casos)

Comprobamos que la entrada fue por pulsación y no por “liberar” la tecla:  
comprobamos que alguna de las líneas RB4 a RB7 está a “0”

IF PARA\_ICD2 == 0

;Si no se va a utilizar ICD2 se ensambla esta parte

;En la interrupción entraremos cuando haya "cambios", tanto si se pulsa como

;si se suelta tecla, por tanto para identificar debemos de comprobar que hay alguna

;pulsada

movlw 0xF0

;Comprobamos los 4 bits mas altos

xorwf PORTB,W

;si estan a uno todos, nos salimos

btfsc STATUS,Z

;para ello comprobamos el bit Z de STATUS

goto SALIDA

;si Z=1 es que estaban los 2 a 1 y salimos

ELSE

;en el caso de que usemos el ICD2 se ensambla el bloque que viene a continuación para

;probar con el ICD2: ya que RB6 y RB7 no están conectadas al teclado y pueden

;tener otros valores distintos

movlw 0x30

;En el caso de que usemos RB6 y RB7 para el ICD2

andwf PORTB,W

;sólo buscamos pulsaciones en RB5 y RB4

xorlw 0x30

;por ese motivo situamos una máscara para

btfsc STATUS,Z

;esos dos bits

goto SALIDA

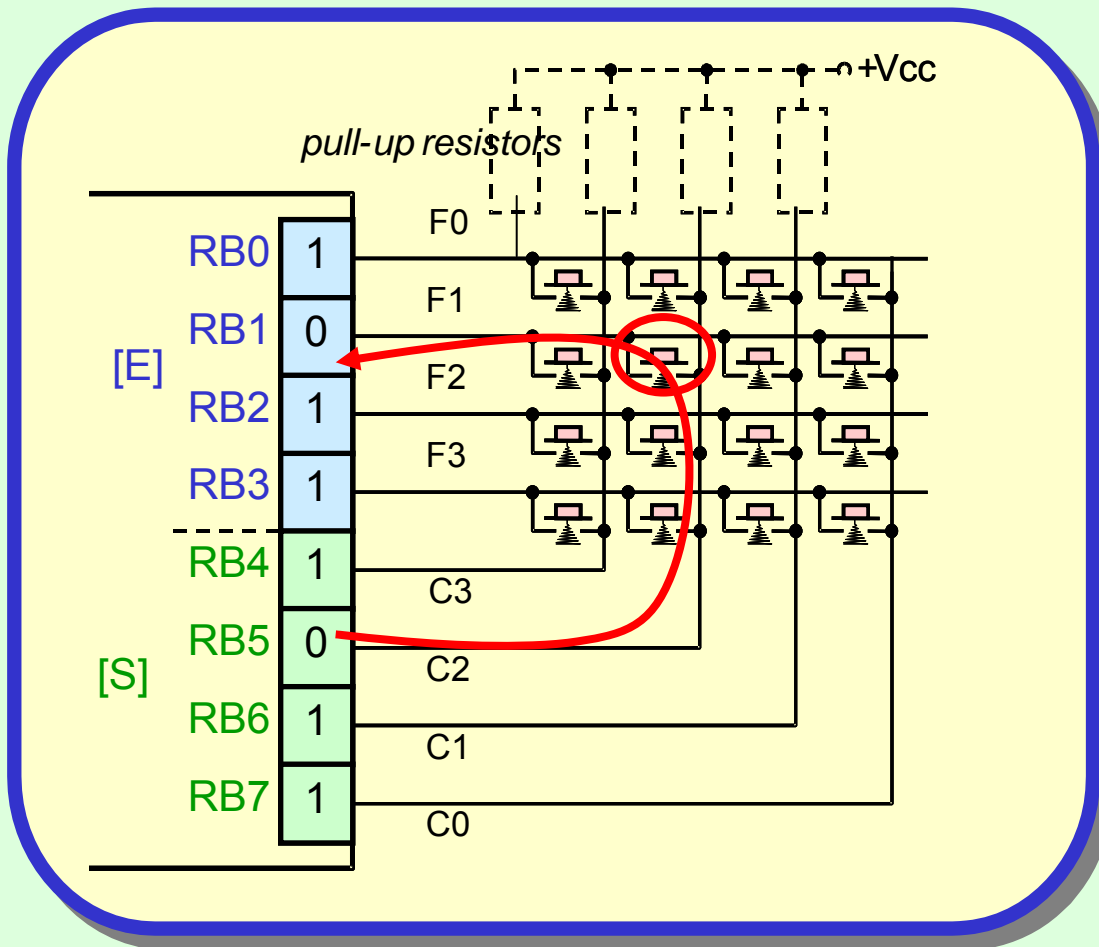
;y si están los dos a '1', nos salimos

ENDIF

;cierre de la directiva de ensamblado condicional

Sentencia de ensamblado condicional, PARA\_ICD2=0 ó distinto de 0,  
en función de que se grabe el código final o se use ICD2 como depurador

Vamos a utilizar la inversión de línea:



Cambiamos el sentido de las líneas del PORTB y sacamos por las nuevas salidas lo que leímos con anterioridad

- Almacenamos los niveles leídos en RB7 a RB4 en la posición LEIDO
- Cambiamos sentido de las líneas
- Sacamos en las nuevas salidas el valor leído

;Almacenamos el PORTB que se ha leído para sacarlo después

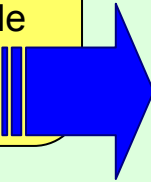
```
movf PORTB,W      ;guardamos PORTB en W
movwf LEIDO        ;y se lo pasamos a LEIDO
```

;Cambiamos entradas y salidas para la inversión de línea:

```
bsf STATUS,RP0     ;Pasamos al banco 1
movlw b'00001111'  ;y ponemos RB4 a RB7 como salidas, RB3 a RB0 entradas
movwf TRISB        ;en el registro de dirección de datos
bcf STATUS,RP0     ;volvemos al banco 0
```

;Sacamos al PORTB la lectura anterior

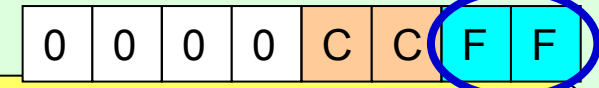
```
movf LEIDO,W      ;Cargamos el valor leído en W
movwf PORTB       ;y ahora va al PORTB
nop               ;Esperamos un par de ciclos para asegurar estabilización de
nop               ;las salidas del PORTB
```





Identificación de la FILA pulsada

CÓDIGO de una TECLA en W



COLUMNA

FILA

;Cada tecla la vamos a codificar en los 4 bits de menos peso de W (16 combinaciones):

;2 bits para la columna (bits 3 y 2) y 2 bits para la fila (bits 1 y 0),

;posteriormente acudiremos a una tabla que nos proporcione la tecla asociada

```

btfss PORTB,0      ;Ahora vamos a buscar el '0' en filas
goto FILA0         ;si RB0=0 es que pertenece a la fila 0
  
```

```

btfss PORTB,1      ;si no, seguimos buscando el '0'
goto FILA1         ;si RB1=0 es que correspondía a la fila 1
  
```

```

btfss PORTB,2      ;si hay un '0' en RB2, se trata de una tecla en la fila 2
goto FILA2         ;y vamos a esa etiqueta
  
```

;Si llegamos aquí es que hay un '0' en RB3 y se trataba de la fila 3

```

goto FILA3         ;y vamos a la parte de esa codificación
  
```

;Vamos a codificar en los 2 últimos bits de W la fila -> W: c1-c0-f1-f0

```

FILA0      clrw      ;Si era la fila 0 cargo dos ceros en los bits bajos de W
  
```

```

goto COLUMNA ;y voy a codificar la columna
  
```

```

FILA1      movlw 0x01 ;si era la fila 1, se carga 01
  
```

```

goto COLUMNA ;y vamos a por la columna
  
```

```

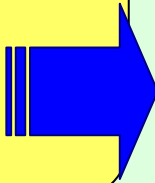
FILA2      movlw 0x02 ;si era la fila 2
  
```

```

goto COLUMNA ;y buscamos la columna
  
```

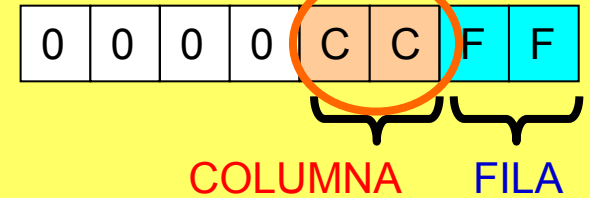
```

FILA3      movlw 0x03 ;finalmente si se trataba de la fila 3
  
```



## Identificación de la COLUMNA pulsada

;Buscamos ahora la codificación de la columna en los bits 3 y 2



```
COLUMNA btfss PORTB,4
        goto COL_3
        btfss PORTB,5
        goto COL_2
```

```
;Si RB4 a '0' es que era la columna 4
;y vamos a completar el código
;Si RB5 es '0' se trataba de la columna 2
;y vamos a por ese código
```

```
btfss PORTB,6
goto COL_1
```

```
;Si RB6 es '0' es que era la columna 1
;y añadimos esa parte de código
```

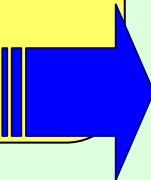
;Si llegamos aquí es que estaba RB7 a '0' y por tanto se trataba de la columna 0

```
COL_0    goto CODIGO
COL_1    iorlw b'00000100'
        goto CODIGO
COL_2    iorlw b'00001000'
        goto CODIGO
COL_3    iorlw b'00001100'
```

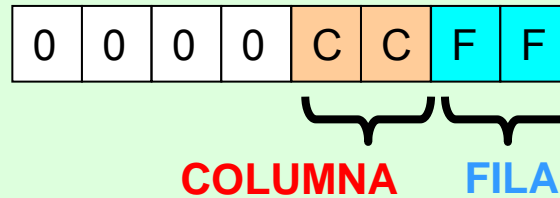
```
;Ya tenemos los bits de columna a 0, nos vamos a por el código
;si se trataba de la columna 1, cargamos 01
;y vamos a por el código
;si era la columna 2, se incuye 10
;y vamos a por el código
;finalmente si era la columna 3, se sitúa 11
```

```
CODIGO    call BUSCA
```

```
;Llamamos a BUSCA llevando en W COLUMNA Y FILA
;y volvemos trayendo en W la tecla asignada a esa fila
;y esa columna
```



TECLA PULSADA	CODIGO COLUMNA (CC)	CODIGO FILA (FF)
7	00	00
4	00	01
1	00	10
A	00	11
8	01	00
5	01	01
2	01	10
0	01	11
9	10	00
6	10	01
3	10	10
B	10	11
F	11	00
E	11	01
D	11	10
C	11	11



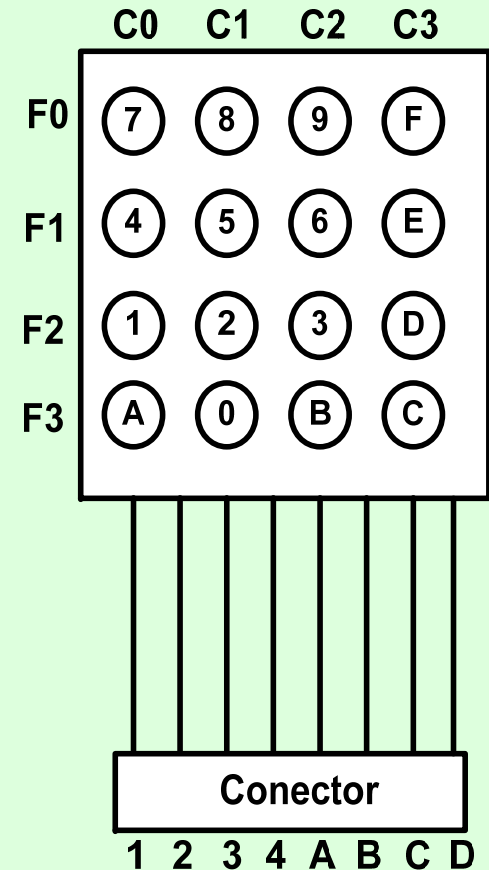
REGISTRO W

## TABLA DE ASIGNACIÓN

BUSCA

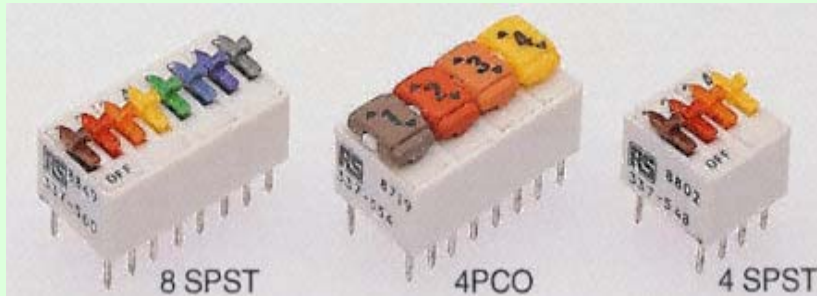
```

addwf PCL,f
retlw 0x07 ;columna 0 fila 0
retlw 0x04 ;columna 0 fila 1
retlw 0x01 ;columna 0 fila 2
retlw 0x0A ;columna 0 fila 3
retlw 0x08 ;columna 1 fila 0
retlw 0x05 ;columna 1 fila 1
retlw 0x02 ;columna 1 fila 2
retlw 0x00 ;columna 1 fila 3
retlw 0x09 ;columna 2 fila 0
retlw 0x06 ;columna 2 fila 1
retlw 0x03 ;columna 2 fila 2
retlw 0x0B ;columna 2 fila 3
retlw 0x0F ;columna 3 fila 0
retlw 0x0E ;columna 3 fila 1
retlw 0x0D ;columna 3 fila 2
retlw 0x0C ;columna 3 fila 3
    
```



## 23.3. Otros dispositivos de entrada

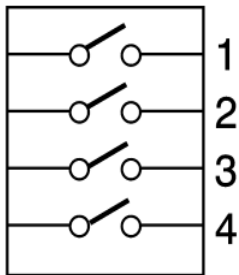
### Otros dispositivos de entrada: MICROINTERRUPTORES



#### Características técnicas típicas

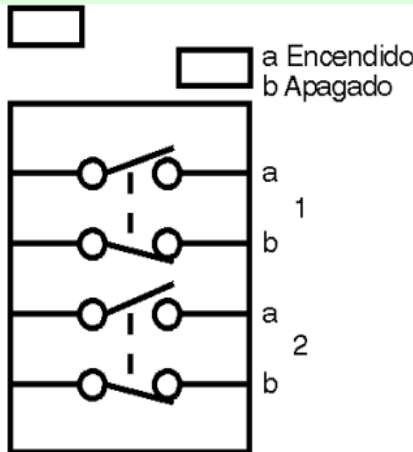
Tipo	Long.	n° pines	Tipo	Long.	n° pines
2 SPST	5	4	SPCO	5	4
4 SPST	10	8	DPCO	10	8
6 SPST	15	12	4PCO	20	16
8 SPST	20	16			
10 SPST	25	20			

Apagado Encendido



Mostrado en el de 4 polos  
(los otros tipos tienen una  
forma similar)

a Apagado  
b Encendido



Mostrado en el de bipolar conmutable  
(los otros tipos tienen una forma similar)

Todos los tipos SPST  
Alt. 8,3 Anch. 10,6

Todos los tipos conmutables  
Alt. 9,2mm Anch. 10,6mm

- Valor nominal de los contactos:  
100 V d.c. 1A continua  
10 VA de conmutación
- Resistencia de contacto: 8mΩ
- Resistencia de aislamiento: 100GΩ a 500V d.c.
- Vida útil mecánica: 20.000 operaciones
- Temperatura de funcionamiento: -55°C a +100°C
- Todos los valores nominales de conmutación de corriente y de potencia citados son para cargas resistivas.

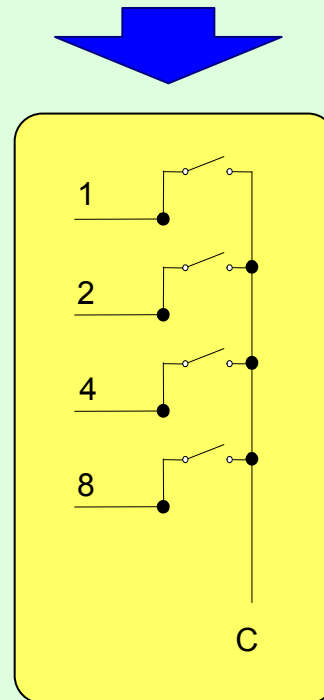
## Preselectores decimales ó interruptores de décadas



Aspecto Físico

Se cierran ó abren los contactos según el número seleccionado en el frontal del preselector

En las líneas  
"8"- "4"- "2"- "1"  
se obtiene el código BCD  
del número seleccionado



Esquema eléctrico

BCD 1-2-4-8; 10 POSITION

WHEEL	SIGNALS CONN. TO COMM. SIG. C				
	1	2	4	8	C
0					●
1	●				●
2		●			●
3	●	●			●
4			●		●
5	●		●		●
6		●	●		●
7	●	●	●		●
8				●	●
9	●			●	●
	1	2	3	4	5
	TERM. LOCATION				

Conexiones entre terminales