Aplicaciones del módulo GPIO: Manejo de teclados matriciales

Sistemas Digitales II -
Microcontroladores
2017-III

Henry Carrillo, Ph.D.

UNIVERSIDAD SERGIO ARBOLEDA

Escuela de Ciencias Exactas e Ingeniería

Contenido – Aplicaciones del módulo GPIO: Manejo de teclados matriciales.

1. **Introducción**
2. **Características**
3. **Uso mediante microcontroladores**
   - **3.1 Serial.**
   - **3.2 Paralelo.**
4. **Manejo de teclados matriciales mediante interrupciones**

Introducción

Teclados Matriciales

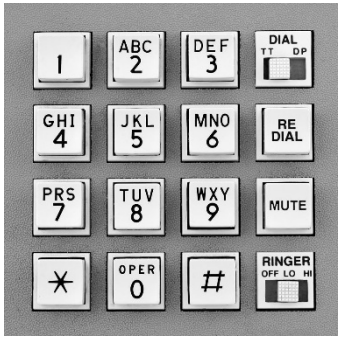
- Que es un teclado matricial? Es un dispositivo mecánico de entrada de datos a sistemas microcontrolados, que contiene un arreglo de N x M pulsadores o interruptores.
  - Típicamente los teclados matriciales usan pulsadores…. Es posible tener sensores capacitivos (táctiles).
  - Los datos son estados lógicos → ALTO o BAJO (Nivel de voltaje definido por el usuario: TTL, CMOS….)

- El tamaño del arreglo es variable → 1 x 4, 3x4, 4x4,…

Teclados Matriciales

- Aplicaciones

Teléfonos fijos



Control de acceso

Cajeros

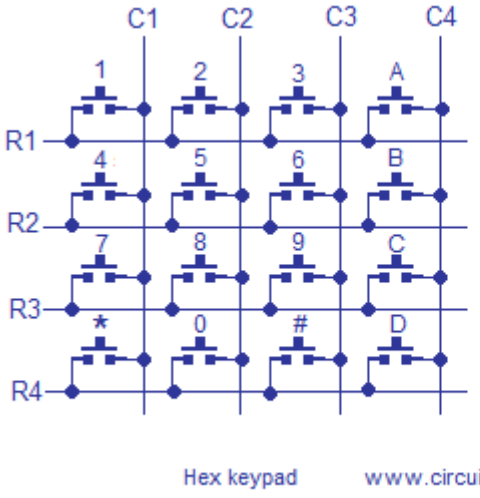
- Usados para ingreso de datos numéricos, alfabéticos o caracteres especiales por parte del usuario.

Características

Teclados Matriciales

- Características I
  - Típicamente los pulsadores se arreglan en filas y columnas → cada pulsador conecta una fila y columna en particular.

- Un teclado matricial tiene dos elementos.
  - Pulsadores o teclas que se designan con una etiqueta visual especifica.
  - Pines de fila y columna → el número de pines es menor a tener pulsadores individuales… con N teclas se necesitan $2\sqrt{(N)}$ líneas de conexión.

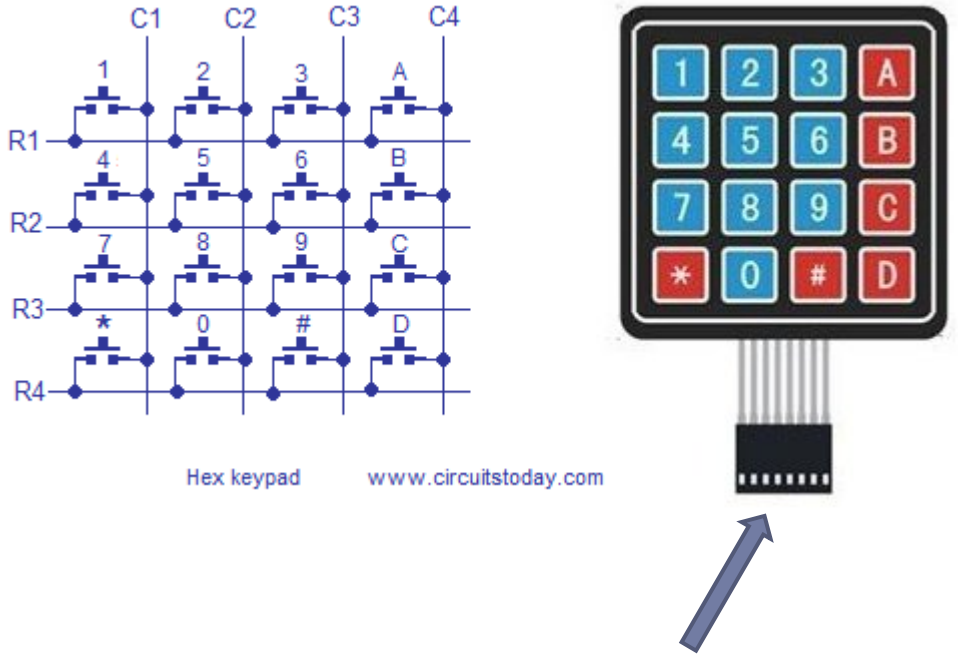


Pines Fila-Columna

Teclados Matriciales

- Características II
  - No necesitan polarización para funcionar → las filas y columnas se interceptan mecánicamente.

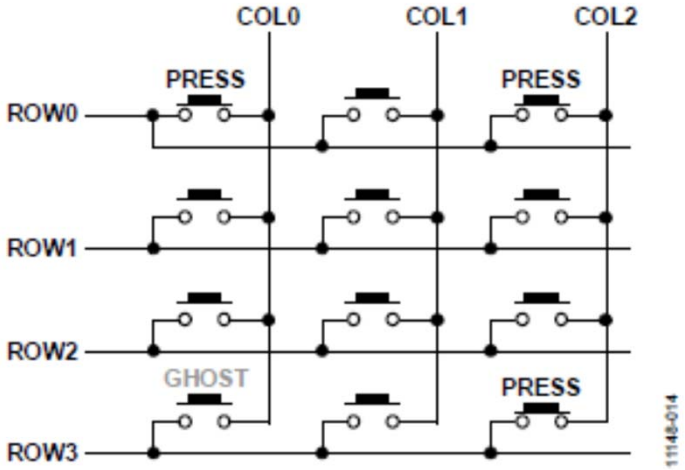
- Al presionar cada tecla muy probablemente se producirán rebotes.
- Se debe garantizar un estado lógico si el pulsador está abierto → Resistencias de *pull-up* y *pull-down*.



Uso mediante microcontroladores

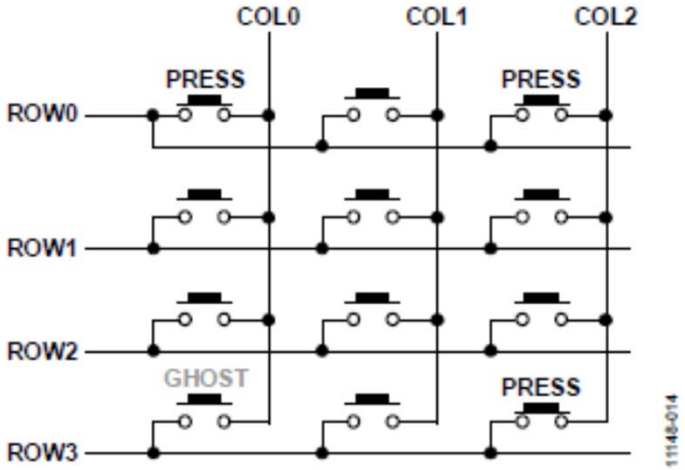
Teclados Matriciales – Lectura mediante microcontrolador

- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila y leer el valor de la columna.
  - Problemas
    - Múltiples teclas conectadas a la misma columna.
    - Valor lógico de las columnas y filas.
- Posibles soluciones
  - **Secuencial**
  - **Paralelo**



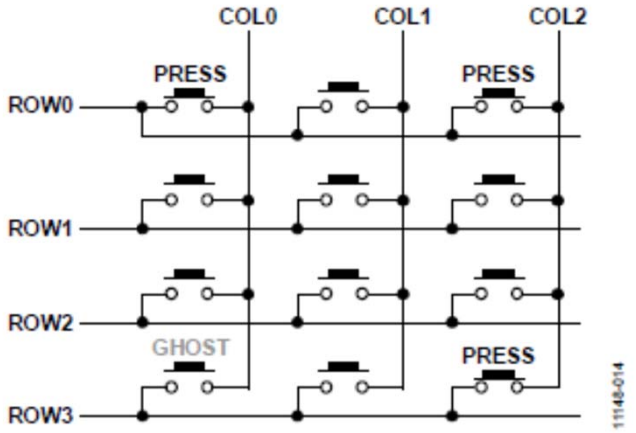
Teclados Matriciales – Lectura mediante microcontrolador

- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila y leer el valor de la columna.
  - Problemas
    - Múltiples teclas conectadas a la misma columna.
    - Valor lógico de las columnas y filas.
- Posibles soluciones
  - **Secuencial**
  - **Paralelo**



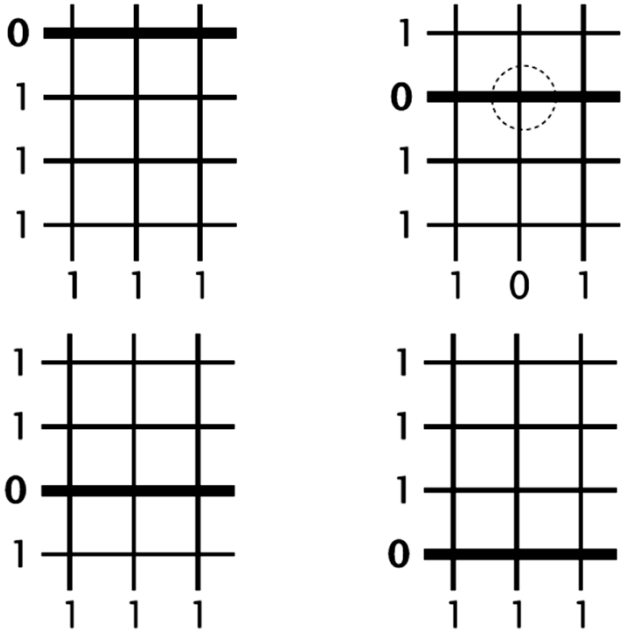
Teclados Matriciales – Lectura mediante microcontrolador "Secuencial"

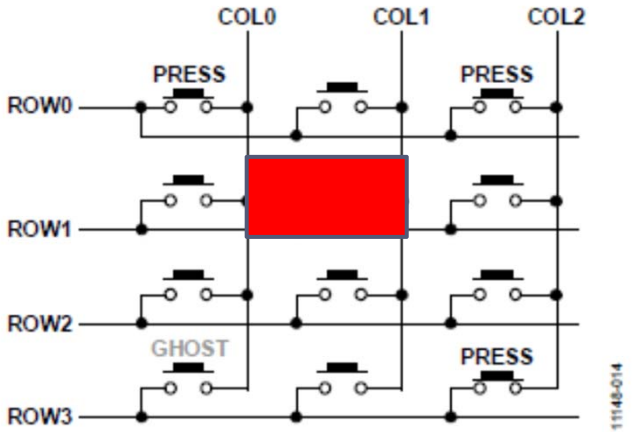
- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila o columna y leer el valor de la columna o fila.
- Una solución secuencial→ Controlar el estado lógico de las filas y verificar el estado de las columnas secuencialmente.
  - Se debe colocar en **ALTO/BAJO** la fila verificada y el resto en **BAJO/ALTO**
  - Se debe monitorear cual columna está en **ALTO/BAJO**
  - Problema : ¿Dos pulsadores de la misma columna activos?
    - Diodos
  - Problemas: ¿Rebotes de los pulsadores?
    - Rutinas de retraso



Teclados Matriciales – Lectura mediante microcontrolador "Secuencial"

- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila o columna y leer el valor de la columna o fila.
- Una solución secuencial→ Controlar el estado lógico de las filas y verificar el estado de las columnas secuencialmente.
  - Se debe colocar en **ALTO/BAJO** la fila verificada y el resto en **BAJO/ALTO**
  - Se debe monitorear cual columna está en **ALTO/BAJO**

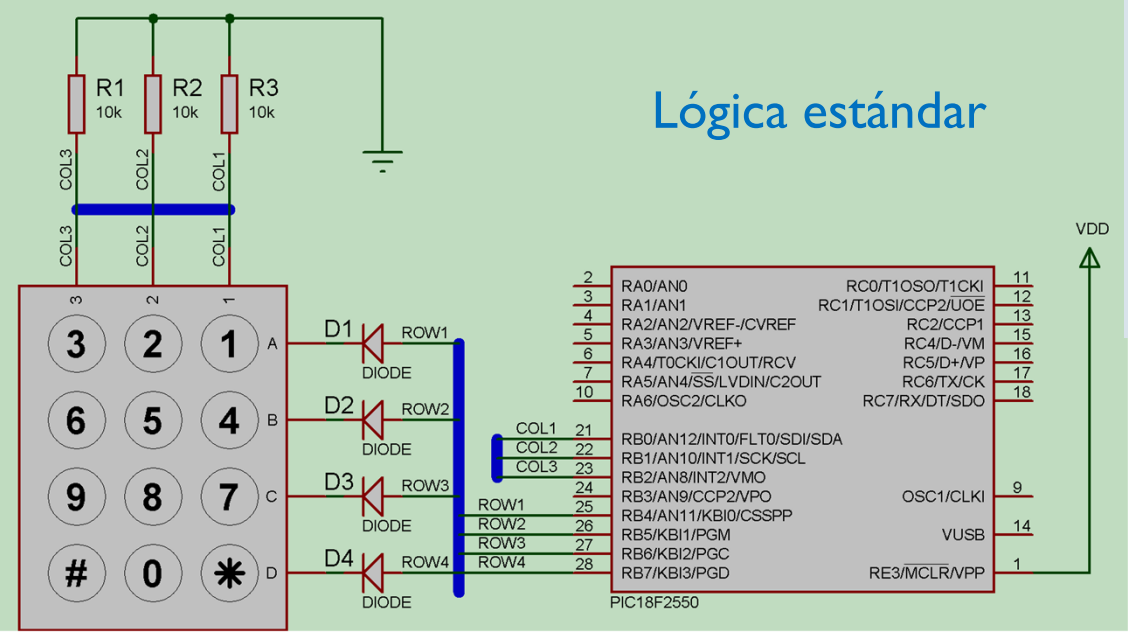




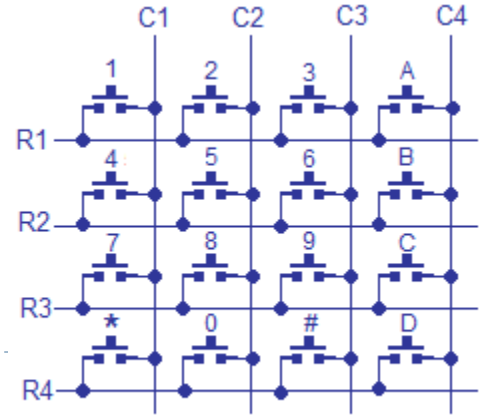
UNIVERSIDAD SERGIO ARBOLEDA

Teclados Matriciales – Interfaz con el microcontrolador "Secuencial"

- Todos los pines del teclado matricial se deben conectar a puertos E/S del microcontrolador
  - El menor entre filas y columnas se conecta como entradas
  - El mayor entre filas y columnas se conecta como salidas

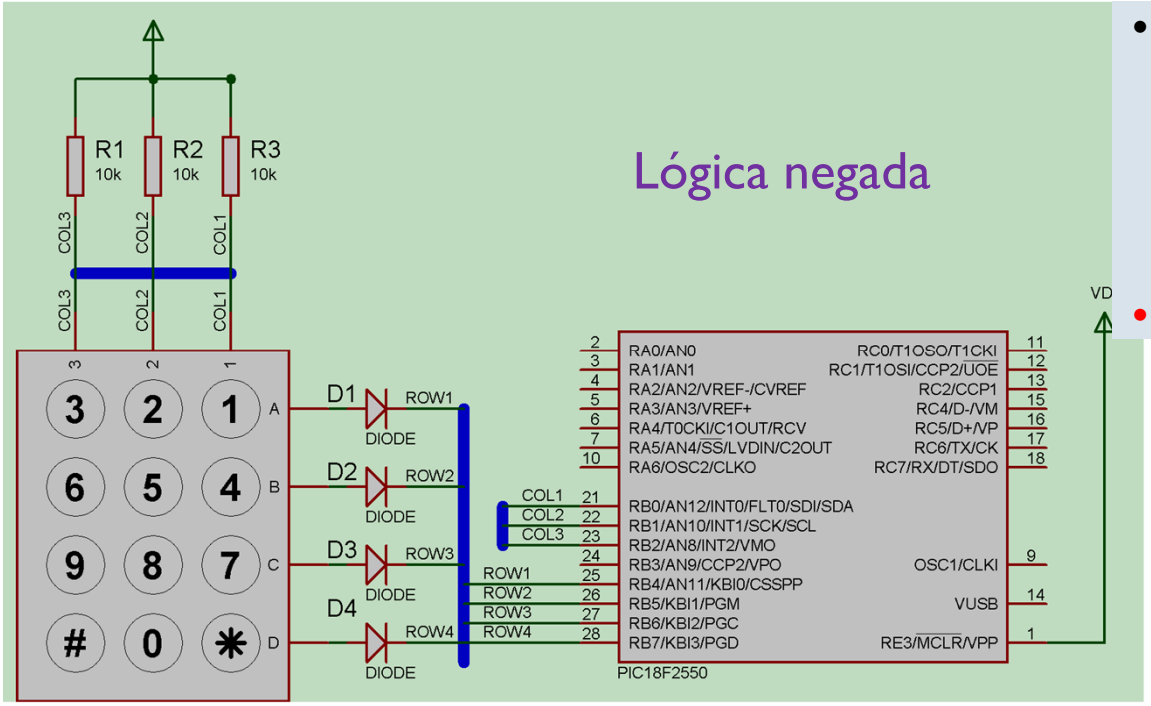


- Se necesita mantener los niveles lógicos → Resistencias de *pull* en los pines de entrada
  - *Pull-up* → Lógica negada
  - *Pull-down* → Lógica estándar
- Diodos

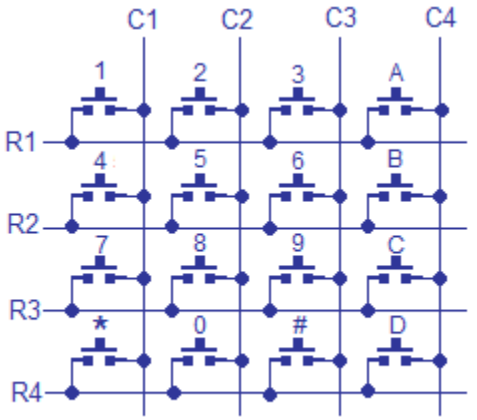


Teclados Matriciales – Interfaz con el microcontrolador "Secuencial"

- Todos los pines del teclado matricial se deben conectar a puertos E/S del microcontrolador
  - El menor entre filas y columnas se conecta como entradas
  - El mayor entre filas y columnas se conecta como salidas



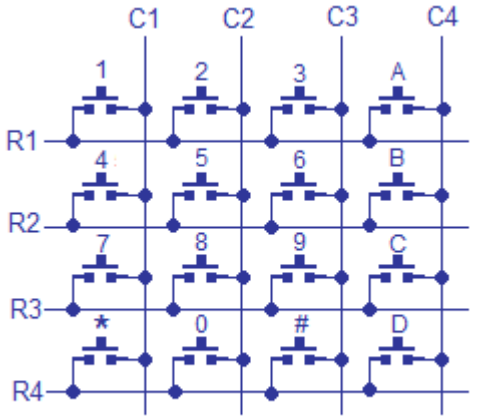
- Se necesita mantener los niveles lógicos → Resistencias de *pull* en los pines de entrada
  - *Pull-up* → Lógica negada
  - *Pull-down* → Lógica estándar
- Diodos



Teclados Matriciales – Lectura mediante microcontrolador "Secuencial"

- Pseudocódigo de una posible implementación

```
void leer_teclado_mat(*RSLT)
{ char found_key = 0;
  *RSLT[0] = *(RSLT+1)= 99; // Condición no encontrado
   for (i = 0; i<row_num; i++)
        { ROW_PORT = BAJO;
          ROW_i = ALTO; // ¿Cómo se implementa ROW_i?
          wait_antirebote ();  //opcional
                for (j=0; j<col_num; j++)
                  {  if (COL_j == ALTO)
                           { *RLST = i;
                             *(RLST+1) = j;
                             found_key = 1;
                             break;}
                  }
          if (found_key == 1) {ROW_PORT = BAJO; return;}
          }
ROW_PORT = BAJO; return;
}
```



UNIVERSIDAD SERGIO ARBOLEDA

Teclados Matriciales – Lectura mediante microcontrolador "Secuencial"

- Ejercicio: Complete el código tal que envíe un ALTO lógico a un pin (0…3) del Puerto C, desde un teclado matricial conectado al puerto B. Solo un Pin debe estar en ALTO al mismo tiempo. Las teclas validas son [1…4]

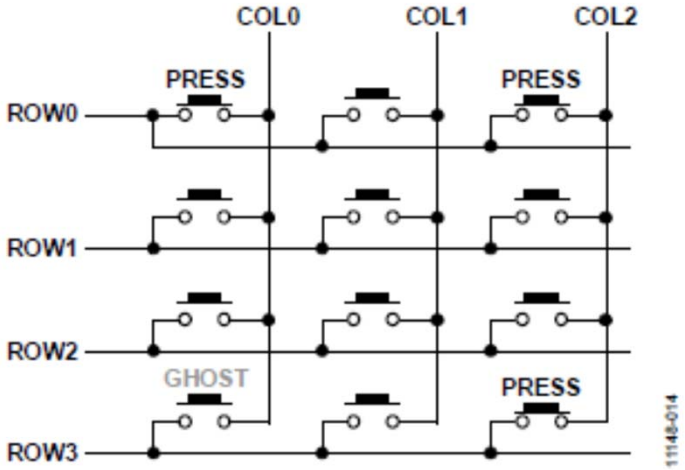
```
#include <xc.h>
#define XTAL_FREQ 4000000
#include <plib/delays.h>

void main()
{
    ****** //Configurar pines
    while(1) //infinite loop
    {
      ****// Leer teclado
      ****// Decodificar teclado y ejecutar tarea
      ****// ¿Está el teclado libre?
    }
}
```

UNIVERSIDAD SERGIO ARBOLEDA

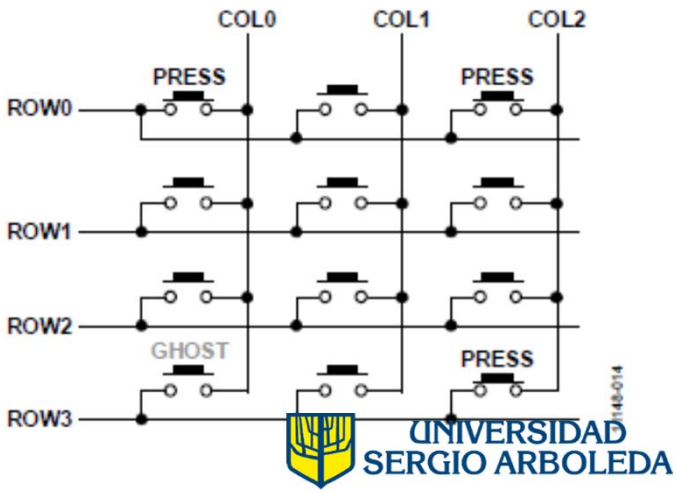
Teclados Matriciales – Lectura mediante microcontrolador

- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila y leer el valor de la columna.
  - Problemas
    - Múltiples teclas conectadas a la misma columna.
    - Valor lógico de las columnas y filas.
- Posibles soluciones
  - **Secuencial**
  - **Paralelo**



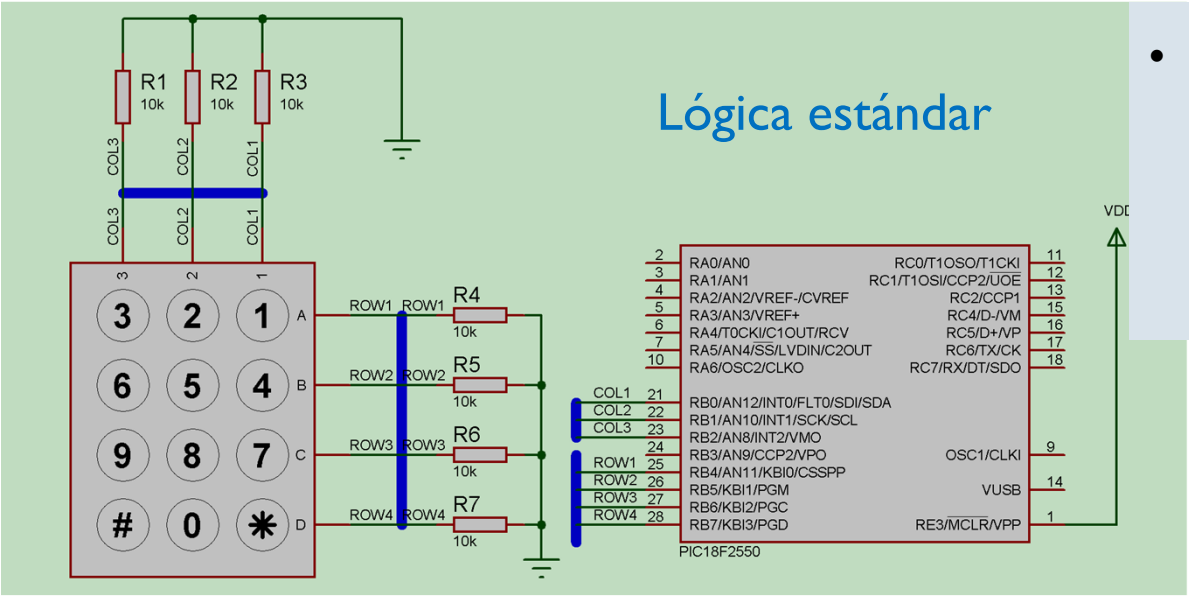
Teclados Matriciales – Lectura mediante microcontrolador "Paralelo"

- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila o columna y leer el valor de la columna o fila.
  - Problema → Múltiples teclas conectadas/apretadas a la misma columna
- Una solución paralela→ activar todas las filas y verificar las columnas. Seguido activar todas las columnas y verificar las filas.
  - Se debe cambiar la dirección de los pines conectados al teclado
  - Se debe guardar los pines en alto en cada verificación
  - Problemas: Rebotes de los pulsadores?
    - Rutinas de retraso

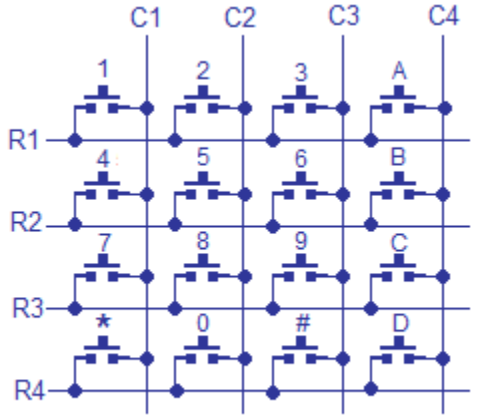


Teclados Matriciales – Interfaz con el microcontrolador "Paralelo"

- Todos los pines del teclado matricial se deben conectar a puertos E/S del microcontrolador
  - Todos los pines necesitan resistencias de *Pull*.
  - No se requieren los diodos.



- Se necesita mantener los niveles lógicos → Resistencias de *pull* en los pines de entrada
  - *Pull-up* → Lógica negada
  - *Pull-down* → Lógica estándar



Teclados Matriciales – Lectura mediante microcontrolador "Paralelo"

- Pseudocódigo de una posible implementación

```
void leer_teclado_mat(* RSLT)
{ char found_key = 0;
*RSLT1 = *(RSLT1+1)= 99; // Condición no encontrado
*RSLT2 = *(RSLT2+1)= 99; // Condición no encontrado
  set_col_input(); // Direcciona los pines de las columnas
  set_row_output(); // Direcciona los pines de las filas
  ROW_PORT = ALTO;
Switch (COL_PORT)
      {
       CASE 0X1F :  *RSLT1 = 0;  break; // Valor depende de los pines conectados
       CASE 0X2F :  *RSLT1 = 1;  break;
       CASE 0X4F :  *RSLT1 = 2;  break;
       Default : *RSLT = 99;
      }
      ROW_PORT = BAJO;
```

UNIVERSIDAD SERGIO ARBOLEDA

Teclados Matriciales – Lectura mediante microcontrolador "Paralelo"

- Pseudocódigo de una posible implementación

```
set_row_input(); // Direcciona los pines de las filas
set_col_output(); // Direcciona los pines de las columnas
 COL_PORT = ALTO;
 Switch (ROW_PORT )
      {CASE 0XF1 :  *(RSLT2+1) = 0;  break; // Valor depende de los pines conectados
       CASE 0XF2 :  *(RSLT2+1) = 1;  break;
       CASE 0XF4 :  *(RSLT2+1) = 2;  break;
       CASE 0XF8 :  *(RSLT2+1) = 3;  break;
       Default : *(RSLT2+1) = 99;}
      COL_PORT = BAJO;
      return;
      }
```

Teclados Matriciales – Lectura mediante microcontrolador "Paralelo"

- Ejercicio: Complete el código tal que un usuario que digite una clave de cuatro dígitos seguido de "*" y esta sea igual a "1234" envíe un ALTO lógico a un pin (RC0) del Puerto C que encenderá un LED. Cualquier clave invalida apagará el LED.

```
#include <xc.h>
#define XTAL_FREQ 4000000
#include <plib/delays.h>
void main()
{
    ****** //Configurar pines
    while(1) //infinite loop
    {
      ****// Leer teclado
      ****// Decodificar teclado y ejecutar tarea
      ****// Está el teclado libre?
    }
}
```

UNIVERSIDAD SERGIO ARBOLEDA

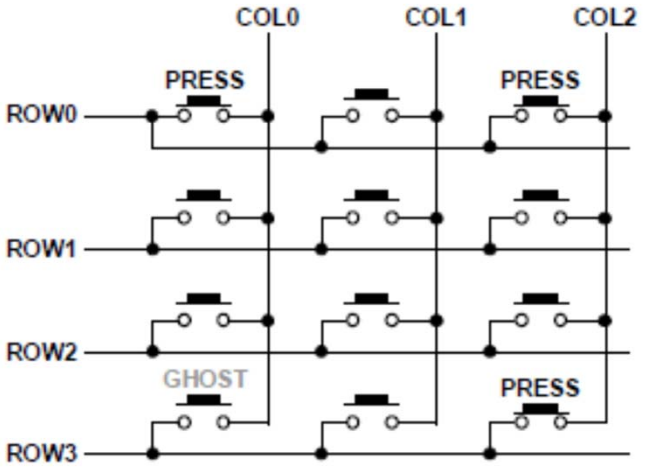
Manejo de teclados matriciales mediante interrupciones

UNIVERSIDAD SERGIO ARBOLEDA

Teclados Matriciales – Lectura mediante microcontrolador

- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila y leer el valor de la columna.
  - Problemas
  - Posibles soluciones: **Secuencial – Paralelo**
    - **Mediante interrupciones → Interrupciones externas**

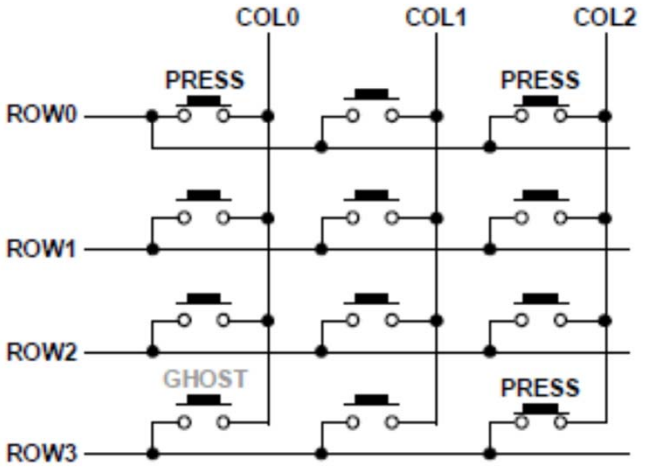
```
#include <xc.h>
#define XTAL_FREQ 4000000
#include <plib/delays.h>
void main()
{
    ****** //Configurar pines
    while(1) //infinite loop
    {
       ****// Leer teclado
       ****// Decodificar teclado y ejecutar tarea
       ****// Está el teclado libre?
    }
}
```



Teclados Matriciales – Lectura mediante microcontrolador

- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila y leer el valor de la columna.
  - Posibles soluciones: **Secuencial – Paralelo**
    - **Mediante interrupciones → Interrupciones externas**
    - Se inicia la secuencia de lectura del teclado una vez se detecte un cambio en el puerto mediante interrupciones….Filas en alto-columnas en bajo

```
#include <xc.h>
#define XTAL_FREQ 4000000
#include <plib/delays.h>
void main()
{ ****** //Configurar pines
    while(1) //infinite loop
    { if (interrupt){
      ****// Leer teclado
      ****// Decodificar teclado y ejecutar tarea
      ****// Está el teclado libre? }
    }
}
```



Mecanismos de Interrupciones – PIC18

- Registros asociados a los mecanismos de interrupción en el PIC18F2550
  - RCON → ***Reset Control Register***
    - Contiene al bit **IPEN** que determina si se utiliza prioridad en las fuentes de interrupción. Además contiene banderas para determinar fuentes de interrupción desde *reset* o modo hibernación.
  - INTCON, INTCON2 e INTCON3 → ***Interrupt Control Register***
    - Permite controlar la habilitación global de las interrupciones y **algunas individuales provenientes de fuentes de interrupción externas.**
  - PIE1 y PIE2 → ***Peripheral Interrupt Enable Registers***
    - Permite controlar individualmente la habilitación de las diferentes fuentes de interrupción, principalmente provenientes de fuentes de interrupción internas.
  - PIR1 y PIR2 → ***Peripheral Interrupt Request Registers***
    - Permite verificar el estado individual de la interrupciones. Cada bit indica el evento de una interrupción en particular.
  - IPR1 e IPR2 → ***Peripheral Interrupt Priority Registers***
    - Permite cambiar individualmente la prioridad de las fuentes de interrupción.

Mecanismos de Interrupciones – PIC18

- Registros asociados a los mecanismos de interrupción en el PIC18

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low-priority peripheral interrupts (if GIE/GIEH = 1)
0 = Disables all low-priority peripheral interrupts

INTCON, INTCON2 e INTCON3
→ *Interrupt Control Register*
Permite controlar la habilitación global de las interrupciones y algunas individuales provenientes de fuentes de interrupción externas

Mecanismos de Interrupciones – PIC18

- Registros asociados a los mecanismos de interrupción en el PIC18

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit(1)
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

INTCON, INTCON2 e INTCON3
→ *Interrupt Control Register*
Permite controlar la habilitación global de las interrupciones y algunas individuales provenientes de fuentes de interrupción externas

Mecanismos de Interrupciones – PIC18

- Registros asociados a los mecanismos de interrupción en el PIC18

REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7							bit 0

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **Unimplemented:** Read as '0'

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **Unimplemented:** Read as '0'

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

INTCON, INTCON2 e INTCON3 → *Interrupt Control Register*
Permite controlar la habilitación global de las interrupciones y algunas individuales provenientes de fuentes de interrupción externas

Mecanismos de Interrupciones – PIC18

- Registros asociados a los mecanismos de interrupción en el PIC18

REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
bit 7							bit 0

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **Unimplemented:** Read as '0'

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 2 **Unimplemented:** Read as '0'

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

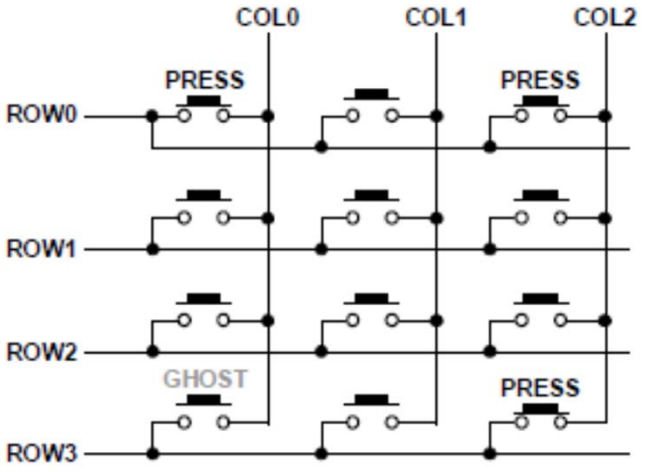
bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

INTCON, INTCON2 e INTCON3
→ *Interrupt Control Register*
Permite controlar la habilitación global de las interrupciones y algunas individuales provenientes de fuentes de interrupción externas

Teclados Matriciales – Lectura mediante microcontrolador

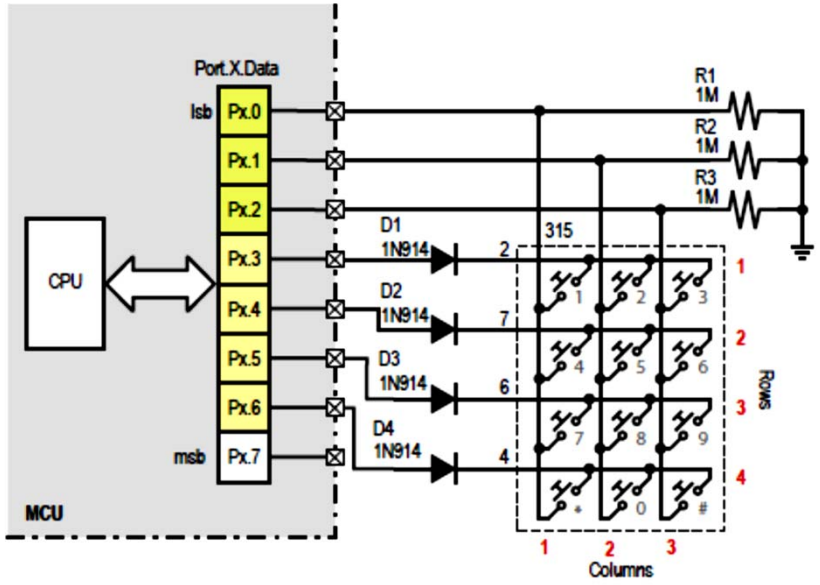
- Idea: Al apretar un pulsador habrá continuidad entre una fila y una columna → Excitar una fila y leer el valor de la columna.
  - Posibles soluciones: **Secuencial – Paralelo**
    - **Mediante interrupciones → Interrupciones externas**
    - Se inicia la secuencia de lectura del teclado una vez se detecte un cambio en el puerto mediante interrupciones.

```
#include <xc.h>
#define XTAL_FREQ 4000000
#include <plib/delays.h>
void main()
{****** //Configurar pines
    while(1) //infinite loop
    { if(interrupt)
       ****// Leer teclado
       ****// Decodificar teclado y ejecutar tarea
       ****// Está el teclado libre?}
    }
}
```



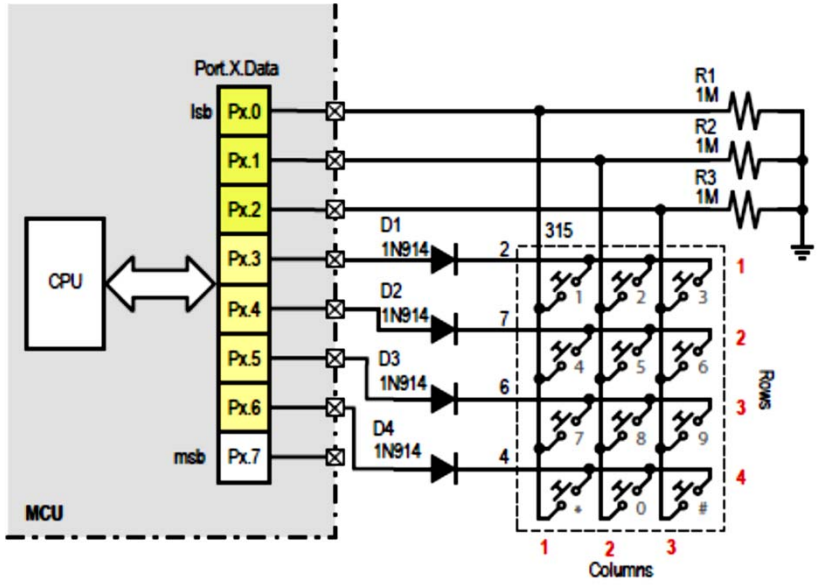
Teclados Matriciales

- Ejercicio: Muestre en un LCD alfanumérico los valores numéricos escritos en un teclado matricial. El valor numérico se mostrará en pantalla una vez se digiten los números (máximo cuatro) y se presione la tecla #. La conexión del teclado al microcontrolador es la dada en la figura.



Teclados Matriciales

- Ejercicio: Muestre en un display 7-segmentos los valores numéricos escritos en un teclado matricial. El valor numérico se mostrará en pantalla una vez se digiten los números (máximo dos) y se presione la tecla #. La conexión del teclado al microcontrolador es la dada en la figura.



Gracias!

Henry Carrillo

http://hcarrillo.co

henry.carrillo@usa.edu.co

Escuela de Ciencias Exactas e Ingeniería

Universidad Sergio Arboleda

Calle 75 No. 15-22 Piso 1°

Bogotá - Colombia

Tel: (+57) (1) 325 7500 ext 2574

UNIVERSIDAD SERGIO ARBOLEDA