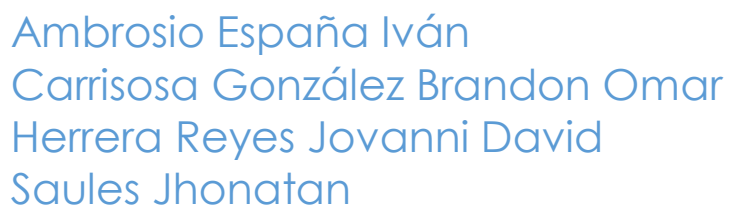


## Microcontroladores.

## Microcontroladores.



Instituto Politécnico Nacional – Escuela Superior de Cómputo.

Profesor: Pérez Pérez José Juan.

## Teclado Matricial.

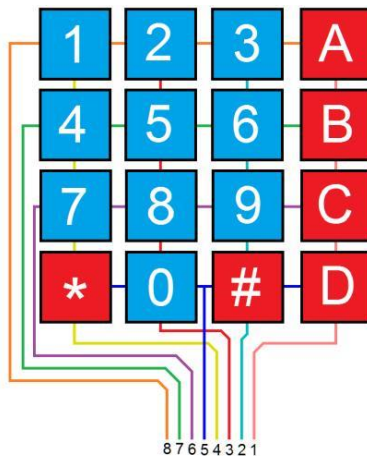
Un teclado matricial es un simple arreglo de botones conectados en filas y columnas, de modo que se pueden leer varios botones con el mínimo número de pines requeridos. Un teclado matricial 4x4 solamente ocupa 4 líneas de un puerto para las filas y otras 4 líneas para las columnas, de este modo se pueden leer 16 teclas utilizando solamente 8 líneas de un microcontrolador. Si asumimos que todas las columnas y filas inicialmente están en alto (1 lógico), la pulsación de un botón se puede detectar al poner cada fila a en bajo (0 lógico) y checar cada columna en busca de un cero, si ninguna columna está en bajo entonces el 0 de las filas se recorre hacia la siguiente y así secuencialmente.

Un modo simple de detectar la tecla presionada es incrementar una variable con la cuenta de las teclas revisadas, de este modo al detectar una pulsación el valor de la cuenta será el valor de la tecla presionada. Si al final no se presionó ninguna tecla la variable se pone a cero y la cuenta vuelve a comenzar. El puerto B del microcontrolador 16f628 (así como en el 16f877) viene preparado especialmente para el control de un teclado matricial 4x4. Para tener siempre un valor de 1 lógico en las columnas del teclado (parte alta del puerto B del pic) es necesario conectar resistencias de pull-up, sin embargo el puerto B cuenta con resistencias de pull-up integradas, de ese modo es posible trabajar con un teclado matricial sin necesidad de ningún componente externo.

Las resistencias de pull-up del puerto B se habilitan poniendo en 0 el bit NOT\_RBPU del registro OPTION\_REG. Al método aquí expuesto para detectar la pulsación de una tecla en un teclado matricial se le conoce como muestreo secuencial. Existen otros, sin embargo éste es tal vez el más sencillo. En el siguiente ejemplo cada tecla del teclado matricial tiene un valor de 0 a F, valor que se despliega en un display de 7 segmentos.

Los Display LCD son visualizadores pasivos, esto significa que no emiten luz como el visualizador o display alfanumérico hecho a base de un arreglo de diodos LEDs. Es por esa razón que, algunas veces, cuando intentamos ver la hora en un reloj que utiliza esta tecnología, es necesario una fuente de luz adicional.

El Display LCD tiene muy bajo consumo de energía si se lo compara con el display o visualizador alfanumérico y son compatibles con la tecnología CMOS, característica que permite que se utilice en equipos portátiles (ejemplo: los relojes de pulsera, calculadoras, etc.). Tiene una vida aproximada de 50,000 horas. Hay diferentes tipos de presentaciones y son muy fáciles de configurar. Hay desde visualizadores comunes de 7 segmentos, hasta una matriz de puntos, todos ellos muy delgados.



### **Código.**

; <http://solocodigo.com/39593/teclado-matricial-y-display-7-segmentos-atmel-avr/>

; Código decodificador de teclado matricial

; Implementado por David Valencia Pesqueira en UPIITA-IPN 2009

; Profesor: Dr. Antonio Hernández Zavala

; GRUPO 4BM3

;

; El siguiente programa decodifica un teclado matricial

; de 16 teclas. Éste tipo de algoritmo está diseñado para

; detectar sólo una tecla a la vez, ya que para detectar

; más de una se requieren diodos por cada interruptor en la  
; construcción del teclado matricial y éso ya queda fuera  
; del enfoque didáctico de éste programa.

;

; Para más información sobre la construcción de teclados  
; matriciales con diodos visitar:

; [http://www.dribin.org/dave/keyboard/one\\_html/](http://www.dribin.org/dave/keyboard/one_html/)

;

;Historial

;v0.1 - Pruebas de las resistencias Pull-UP --- OK

;v0.2 - Probando la salida del tempSCAN

;v0.3 - Nueva estructura de tempSCAN -- OK

;v0.4 - Salida por tira de leds en PORTA -- OK

;

;Hardware en un ATMEL ATmega 8535

;El teclado está conectado de la siguiente manera:

;

;     COLUMNAS

;     0   1   2   3

;

; FILAS 0 X   X   X   X PC4

;

;     1 X   X   X   X PC5

;

;     2 X   X   X   X PC6

;

;     3 X   X   X   X PC7

;

```

;      PC0 PC1 PC2 PC3
;
;#####

;CABECERA
.include "m8535def.inc"

;#####

;Vector de Interrupciones
.org 0x00
    rjmp reset ;interrupción de Reset
;
;#####

;Variables Globales
;La variable tempscan se acomoda de la siguiente forma

;tempscan = [fila3,fila2,fila1,fila0,col3,col2,col1,col0]
;Son "banderas" que se prenden para indicar en qué fila
; y qué columna se halla la tecla.

.def tempscan=r17
.def temp=r16

;Rutina Principal
reset:
    ;inicializar pila
    cli
    ldi temp,low(ramend)

```

```
out spl,temp
ldi temp,high(ramend)
out sph,temp
sei
rjmp init
```

init:

```
;Inicializar Puertos
;Este código emplea el puerto C, con los bits 0 al 3 para las columnas
;y los bits 4 al 7 para las filas.
;Inicia con las filas como salidas con ceros y las columnas como entradas
;que se mantienen en un nivel alto mientras no estén conectadas
;Para lograr esto, se habilitan las "resistencias pull-up" que conectan
;el pin deseado hacia VCC.
;Para hacerlo se declaran los pines como entradas (DDRC=0) pero
;enviamos 0xFF al registro PORTC para activarlas.

;Declarando al nibble alto como salida y al bajo como entrada
ldi temp,0b11110000
out DDRC,temp
;Activando las resistencias pull-up en la parte baja
ldi temp,0x0F
out PORTC,temp
;Si se presiona una tecla, el "0" de alguna fila será
;detectado por las columnas, y entonces comienza el
;proceso de decodificación.

ldi temp,0b10000000
```

```
out DDRD,temp
;Puerto D como salida para cuando detecta tecla
ser temp
out DDRA,temp
;Nibble bajo del puerto A como salida
clr temp
out PORTA,temp
```

```
clr tempscan ;Borra el resultado del escaneo
```

```
rjmp begin
```

```
begin:
```

```
clr temp
out PORTD,temp
;Probar si algun botón fue presionado
in temp,PINC
cpi temp,0x0F ;Si hay algun cero en las columnas, buscar la tecla
brne antirrebotes
rjmp begin
```

```
antirrebotes:
```

```
;Esta rutina espera un corto periodo de tiempo mientras el interruptor
;se estabiliza para evitar que el micro detecte la misma tecla varias
;veces al igual que posible ruido.
;Vuelve a leer las entradas
ser temp
out PORTD,temp
```

```
rcall delay
in temp,PINC
cpi temp,0x0F
;Si detecta el cero de nuevo, procesa la detección
brne deteccion
;Si fue un falso contacto, vuelve a empezar
rjmp begin
;FIN DE ANTIRREBOTES
```

deteccion:

```
clr tempscan
;inicializamos el código de escaneo temporal
ldi tempscan,0x00
;Buscar las columnas
```

colscan:

```
sbis pinc,0 ;prueba la columna 0
ldi tempscan,0x01
sbis pinc,1 ;prueba la columna 1
ldi tempscan,0x02
sbis pinc,2 ;prueba la columna 2
ldi tempscan,0x04
sbis pinc,3 ;prueba la columna 3
ldi tempscan,0x08
```

```
cpi tempscan,0x00 ;si no halló la columna, repetir
breq init
```



;Ahora que se encontró la columna, tienen que invertirse  
;los sentidos del nibble bajo y el alto para buscar la  
;fila  
rcall invertir  
;Ahora invertidos, buscaremos si alguna fila está  
;baja  
in temp,PINC  
cpi temp,0xF0 ;Si hay algun cero en las columnas, buscar la tecla  
brne filadetectada  
rjmp init ;Si no hay fila con cero, devuelve los puertos  
;a su estado original y reinicia el programa

filadetectada:

;Buscar la fila

filascan:

ldi temp,0x10

sbis pinc,4 ;Prueba la fila 0

rjmp col0

sbis pinc,5 ;Prueba la fila 1

rjmp col1

sbis pinc,6 ;Prueba la fila 2

rjmp col2

sbis pinc,7 ;Prueba la fila 3

rjmp col3

rjmp init ;Si no halla la fila, reinicia

;A continuación se escribirá el valor correspondiente

;a la fila, se usa el comando ROL para multiplicar por 2  
;pero se limpia antes la bandera de acarreo para evitar  
;que se sume y entregue valores erróneos.

col0:

add tempscan,temp  
rjmp fincol

col1:

ldi temp,0x20  
add tempscan,temp  
rjmp fincol

col2:

ldi temp,0x40  
add tempscan,temp  
rjmp fincol

col3:

ldi temp,0x80  
add tempscan,temp  
rjmp fincol

fincol:

;Toca procesar el código temporal para sacarlo  
;por los leds, en este caso sale en 4 leds en puertoA  
out PORTA,tempscan  
clr tempscan

rcall desinvertir ;devolver los puertos al estado original

rjmp begin

;Subrutinas

delay:

cli

;Respalda variables globales en la pila

push r16

push r17

;Inicia secuencia de retraso

loop0: ldi r17,0x04

loop1: ldi r16,0xFF

loop2: dec r16

brne loop2

dec r17

brne loop1

sei

;Restaurar las variables desde la pila

pop r17

pop r16

ret

invertir: ;Inversión de los pines

ldi temp,0b00001111

out DDRC,temp

;Activando las resistencias pull-up en la parte alta

ldi temp,0xF0

```
out PORTC,temp
;Ahora la parte alta es entrada con pullup y la parte
;baja son salidas con ceros
ret
```

```
desinvertir: ;Devolver los pines al estado inicial
ldi temp,0b11110000
out DDRC,temp
;Activando las resistencias pull-up en la parte alta
ldi temp,0x0F
out PORTC,temp
;Ahora la parte baja es entrada con pullup y la parte
;alta son salidas con ceros
ret
```

```
;FIN DEL PROGRAMA TECLADO MATRICIAL
```

## **Simulación.**

## **Conclusión.**

El principio de funcionamiento es sencillo y es similar a lo que haríamos para multiplexar leds o displays de 7 segmentos. El programa configura el puerto B del PIC de la siguiente forma: RB4 a RB7 funcionan como salidas y la otra mitad (RB0-RB3) como entradas. Las filas (horizontal) del teclado matricial se conectan a los bits más significativos que funcionan como salidas, mientras que las columnas (vertical) se conectan a los bits menos significativos del puerto que funcionan como entradas con resistencias pull-down. Cualquier tecla que se oprima en una columna causará que uno de los bits menos significativos del puerto (RB0 – RB3) cambie de un estado lógico bajo a un estado alto.

El microcontrolador escanea en forma sucesiva los pines de salida, mientras lee las entradas en la parte baja del puerto, de manera que puede detectar que teclas están oprimidas en cada fila. Ahora solo falta escribir nuestro código en C que implemente los procedimientos mencionados anteriormente y nos devuelva un valor de la tecla oprimida, por ejemplo, mediante un número binario.

## **Referencias.**

[www.circuitoselectronicos.org/2011/03/teclado-matricial-4x4.html](http://www.circuitoselectronicos.org/2011/03/teclado-matricial-4x4.html)

<http://www.geekfactory.mx/tutoriales/tutoriales-pic/teclado-matricial-con-pic/>

[www.sites.upiicsa.ipn.mx/polilibros/portal/polilibros/p...V/Unidad%20V\\_2.htm](http://www.sites.upiicsa.ipn.mx/polilibros/portal/polilibros/p...V/Unidad%20V_2.htm)