

Teclado 4x4™

Manual de
usuario

Todos los sistemas de desarrollo de Mikroelektronika disponen de un gran número de módulos periféricos, ampliando el rango de aplicaciones de los microcontroladores y facilitando el proceso de probar los programas. Aparte de estos módulos, es posible utilizar un gran número de módulos adicionales conectados al sistema de desarrollo por los conectores de puertos de E/S. Algunos de estos módulos adicionales pueden funcionar como dispositivos autónomos sin estar conectados al microcontrolador.

Placa adicional

 **MikroElektronika**

SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

Teclado 4x4

El *teclado 4x4* se utiliza para introducir los datos numéricos en el microcontrolador. Consiste en 16 botones organizados en forma de matriz en cuatro filas y cuatro columnas. Está conectado al sistema de desarrollo por un conector hembra IDC 10 regular conectado a un puerto del sistema de desarrollo.

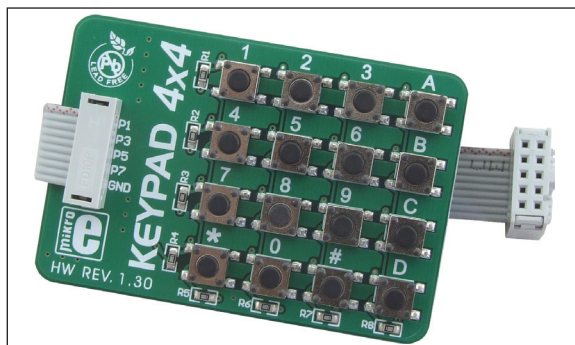


Figura 1: Teclado 4x4

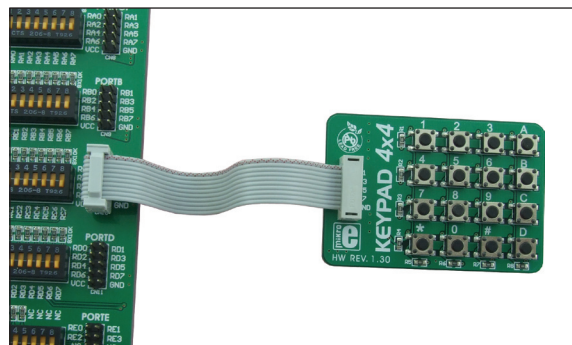


Figura 2: Teclado 4x4 conectado al sistema de desarrollo

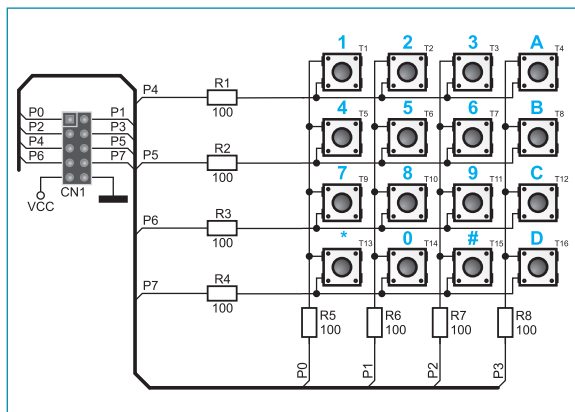


Figura 3: Esquema de conexión del Teclado 4x4

El teclado se utiliza de la siguiente manera:

1. Cuatro pines del microcontrolador se deben configurar como salidas, mientras que los otros cuatro pines se deben configurar como entradas. Para que el teclado funcione apropiadamente, las resistencias pull-down se deben conectar a los pines de entrada del microcontrolador. Así se define el estado lógico sin pulsar ningún botón.
2. Luego, los pines de salida están a uno lógico (1), mientras que se lee el estado lógico de los pines de entrada. Al pulsar un botón, un uno lógico (1) aparecerá en algún pin de entrada.
3. Al combinar unos y ceros en los pines de salida, se define cuál botón fue pulsado.

Una manera más fácil de introducir los datos al utilizar el teclado 4x4 es por medio de utilizar las funciones listas para ser utilizadas que se encuentran en la librería *Keypad Library* de cualquier compilador de Mikroelektronika. En las siguientes páginas hay tres simples ejemplos escritos para el microcontrolador PIC16F887 en los lenguajes de programación *mikroC*, *mikroBasic* y *mikroPascal*. En todos los ejemplos, el número introducido por medio del teclado se convierte en el código ASCII equivalente (0...9, A...F) y luego se visualiza en la segunda fila en el LCD. En este caso, las resistencias pull-down se conectan a los pines de salida RD0 - RD3 y un cero (0) lógico aparecerá en estos pines en el estado inactivo.

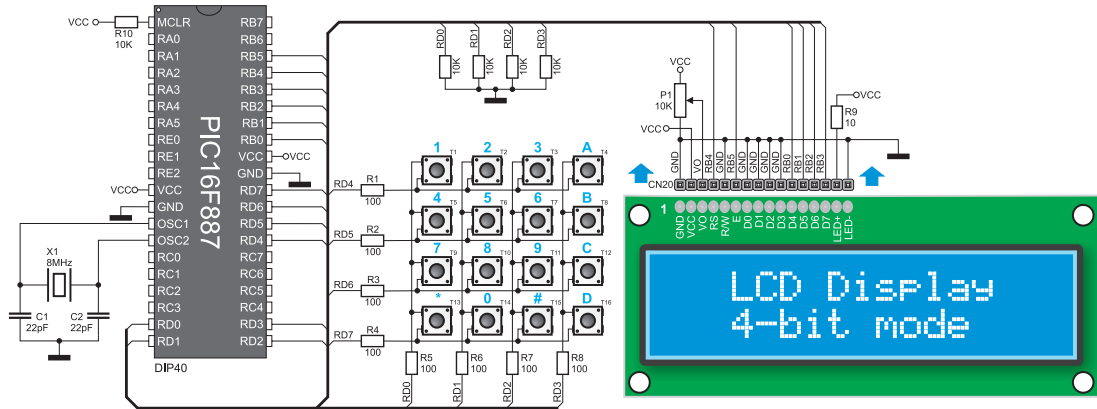


Figura 4: Esquema de conexión entre el teclado, LCD y el microcontrolador

Ejemplo 1: Programa escrito en mikroC PRO for PIC

```

unsigned short kp, cnt, oldstate = 0;
char txt[6];

char keypadPort at PORTD;

sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;

void main() {
    cnt = 0;
    Keypad_Init();
    ANSEL = 0;
    ANSELH = 0;
    Lcd_Init();
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Cmd(LCD_CURSOR_OFF);
    Lcd_Out(1, 1, "1");
    Lcd_Out(1, 1, "Key :");
    Lcd_Out(2, 1, "Times:");

    do {
        kp = 0;

        // Esperar que la tecla se presione y suelte
        do
            kp = Keypad_Key_Click();

        while (!kp);

        // Preparar el valor para la salida, convertir el dato en el código ASCII equivalente

        switch (kp) {
            //case 10: kp = 42; break; // '*'
            //case 11: kp = 48; break; // '0'
            //case 12: kp = 35; break; // '#'

            //default: kp += 48;
            case 1: kp = 49; break; // 1
            case 2: kp = 50; break; // 2
            case 3: kp = 51; break; // 3
            case 4: kp = 65; break; // A
            case 5: kp = 52; break; // 4
            case 6: kp = 53; break; // 5
            case 7: kp = 54; break; // 6
            case 8: kp = 66; break; // B
            case 9: kp = 55; break; // 7
            case 10: kp = 56; break; // 8
            case 11: kp = 57; break; // 9
            case 12: kp = 67; break; // C
            case 13: kp = 42; break; // *
            case 14: kp = 48; break; // 0
            case 15: kp = 35; break; // #
            case 16: kp = 68; break; // D
        }

        if (kp != oldstate) {
            cnt = 1;
            oldstate = kp;
        }
        else {
            cnt++;
        }

        Lcd_Chrc(1, 10, kp);

        if (cnt == 255) {
            cnt = 0;
            Lcd_Out(2, 10, " ");
        }

        WordToStr(cnt, txt);
        Lcd_Out(2, 10, txt);
    } while (1);
}

```

Ejemplo 2: Programa escrito en mikroBasic PRO for PIC

```

program Keypad_Test
dim kp, cnt, oldstate as byte
txt as char[7]

' Conexiones del módulo teclado
dim keypadPort as byte at PORTD
' Final de conexiones del módulo teclado

' Conexiones del módulo LCD
dim LCD_RS as sbit at RB4_bit
LCD_EN as sbit at RB5_bit
LCD_D4 as sbit at RB0_bit
LCD_D5 as sbit at RB1_bit
LCD_D6 as sbit at RB2_bit
LCD_D7 as sbit at RB3_bit

LCD_RS_Direction as sbit at TRISB4_bit
LCD_EN_Direction as sbit at TRISB5_bit
LCD_D4_Direction as sbit at TRISB0_bit
LCD_D5_Direction as sbit at TRISB1_bit
LCD_D6_Direction as sbit at TRISB2_bit
LCD_D7_Direction as sbit at TRISB3_bit
' Final de conexiones del módulo LCD

main:
oldstate = 0
cnt = 0 ' Reiniciar el contador
Keypad_Init() ' Inicializar el teclado
ANSEL = 0 ' Configurar los pines AN como digitales de E/S
ANSELH = 0
Lcd_Init() ' Inicializar el LCD
Lcd_Cmd(_LCD_CLEAR) ' Borrar el visualizador
Lcd_Cmd(_LCD_CURSOR_OFF) ' Cursor apagado
Lcd_Out(1, 1, "Key :") ' Escribir el texto del mensaje en el LCD
Lcd_Out(2, 1, "Times:")

while TRUE

kp = 0 ' Reiniciar la variable del código de tecla

' Esperar que la tecla se presione y suelte
while ( kp = 0 )
kp = Keypad_Key_Click() ' Almacenar el código de la tecla en la variable kp

wend
' Preparar el valor para la salida, convertir el dato en el código ASCII equivalente
select case kp
'case 10: kp = 42 ' ""
'case 11: kp = 48 ' "0"
'case 12: kp = 35 ' "#"
'default: kp += 48

```

```

case 1
kp = 49 ' 1 ' No comentar este bloque para el teclado 4x4
case 2
kp = 50 ' 2
case 3
kp = 51 ' 3
case 4
kp = 65 ' A
case 5
kp = 52 ' 4
case 6
kp = 53 ' 5
case 7
kp = 54 ' 6
case 8
kp = 66 ' B
case 9
kp = 55 ' 7
case 10
kp = 56 ' 8
case 11
kp = 57 ' 9
case 12
kp = 67 ' C
case 13
kp = 42 ' *
case 14
kp = 48 ' 0
case 15
kp = 35 ' #
case 16
kp = 68 ' D

end select

if (kp <> oldstate) then ' Tecla presionada difiere de la anterior
cnt = 1
oldstate = kp
else ' Tecla presionada es igual que la anterior
Inc(cnt)
end if
Lcd_Chr(1, 10, kp) ' Visualizar el valor ASCII del dato en el LCD
if (cnt = 255) then ' Si la variable del contador está fuera del rango
cnt = 0
Lcd_Out(2, 10, " ")
end if

WordToStr(cnt, txt) ' Convertir el valor del contador en la cadena
Lcd_Out(2, 10, txt) ' Visualizar el valor del contador en el LCD
wend
end.

```

Ejemplo 3: Programa escrito en mikroPascal PRO for PIC

```
program Keypad_Test;

var kp, cnt, oldstate : byte;
    txt : array[6] of byte;

// Conexiones del módulo teclado
var keypadPort : byte at PORTD;
// Final de conexiones del módulo teclado

// Conexiones del módulo LCD
var LCD_RS : sbit at RB4_bit;
    LCD_EN : sbit at RB5_bit;
    LCD_D4 : sbit at RB0_bit;
    LCD_D5 : sbit at RB1_bit;
    LCD_D6 : sbit at RB2_bit;
    LCD_D7 : sbit at RB3_bit;

var LCD_RS_Direction : sbit at TRISB4_bit;
    LCD_EN_Direction : sbit at TRISB5_bit;
    LCD_D4_Direction : sbit at TRISB0_bit;
    LCD_D5_Direction : sbit at TRISB1_bit;
    LCD_D6_Direction : sbit at TRISB2_bit;
    LCD_D7_Direction : sbit at TRISB3_bit;
// Final de conexiones del módulo LCD

begin
oldstate := 0;
cnt := 0; // Reiniciar el contador
Keypad_Init(); // Inicializar el teclado
ANSEL := 0; // Configurar los pines AN como digitales de E/S
ANSELH := 0;
Lcd_Init(); // Inicializar el LCD
Lcd_Cmd(_LCD_CLEAR); // Borrar el visualizador
Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor apagado
Lcd_Out(1, 1, 'Key :'); // Escribir el texto del mensaje en el LCD
Lcd_Out(2, 1, 'Times:');

while TRUE do
begin
kp := 0; // Reiniciar la variable del código de tecla

// Esperar que la tecla se presione y suelte
while ( kp = 0 ) do
kp := Keypad_Key_Click(); // Almacenar el código de la tecla en la variable kp
// Preparar el valor para la salida, convertir el dato en el código ASCII equivalente
case kp of
//case 10: kp = 42; // '**' // No comentar este bloque para el teclado4x3
//case 11: kp = 48; // '0'
//case 12: kp = 35; // '#'
//default: kp += 48;

1: kp := 49; // 1 // No comentar este bloque para el teclado 4x4
2: kp := 50; // 2
3: kp := 51; // 3
4: kp := 65; // A
5: kp := 52; // 4
6: kp := 53; // 5
7: kp := 54; // 6
8: kp := 66; // B
9: kp := 55; // 7
10: kp := 56; // 8
11: kp := 57; // 9
12: kp := 67; // C
13: kp := 42; // *
14: kp := 48; // 0
15: kp := 35; // #
16: kp := 68; // D

end;

if (kp <> oldstate) then // Tecla presionada difiere de la anterior
begin
cnt := 1;
oldstate := kp;
end
else // Tecla presionada es igual que la anterior
Inc(cnt);

Lcd_Chr(1, 10, kp); // Visualizar el valor ASCII del dato en el LCD

if (cnt = 255) then // Si la variable del contador está fuera del rango
begin
cnt := 0;
Lcd_Out(2, 10, ' ');
end;

WordToStr(cnt, txt); // Convertir el valor del contador en la cadena

Lcd_Out(2, 10, txt); // Visualizar el valor del contador en el LCD
end;
end.
```