

MikroElektronika books

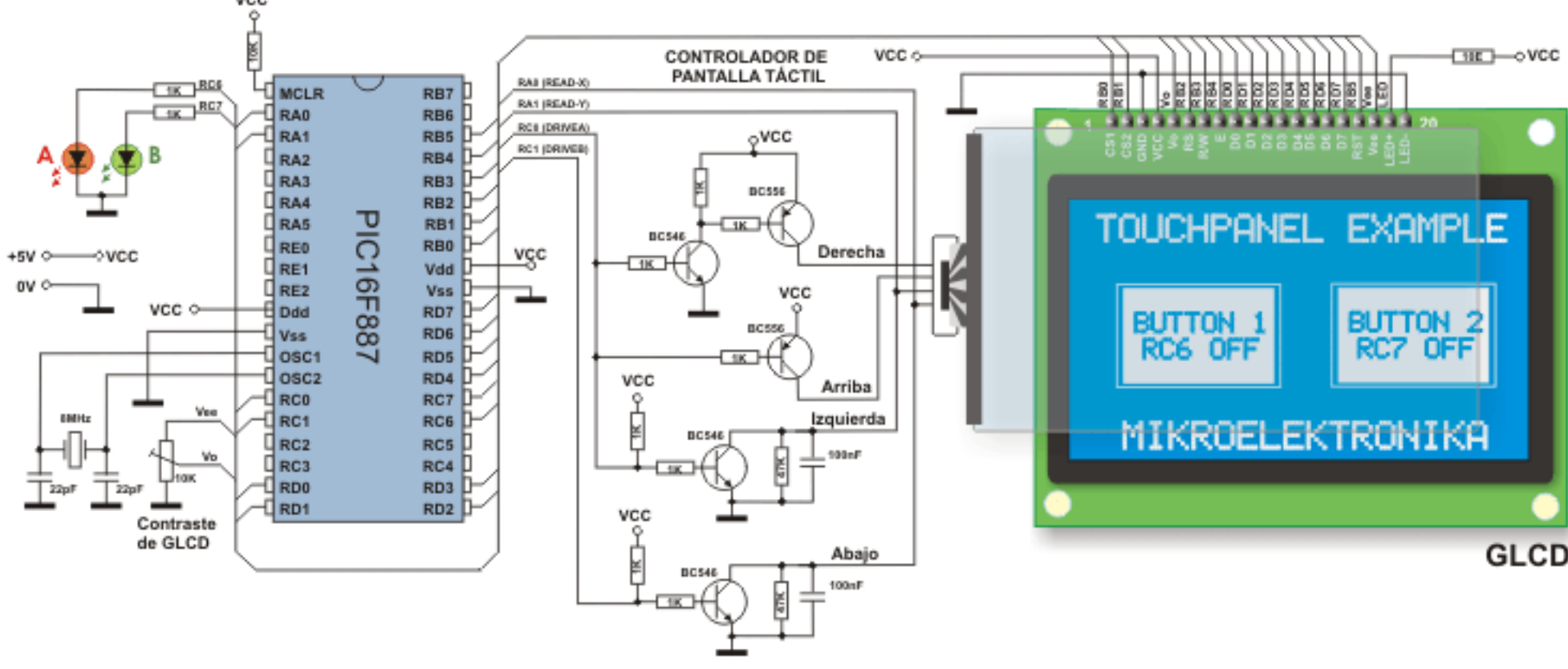
Book: [Microcontroladores PIC – Programación en C con ejemplos](#)

Table of Contents

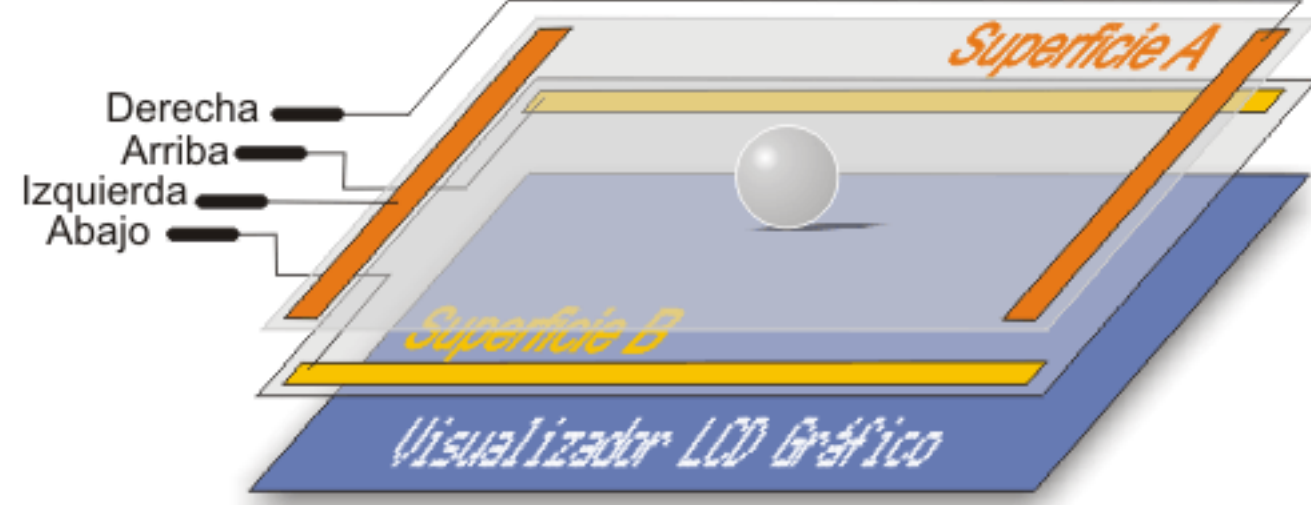
4.17 Ejemplo 15

Utilizar el panel táctil...

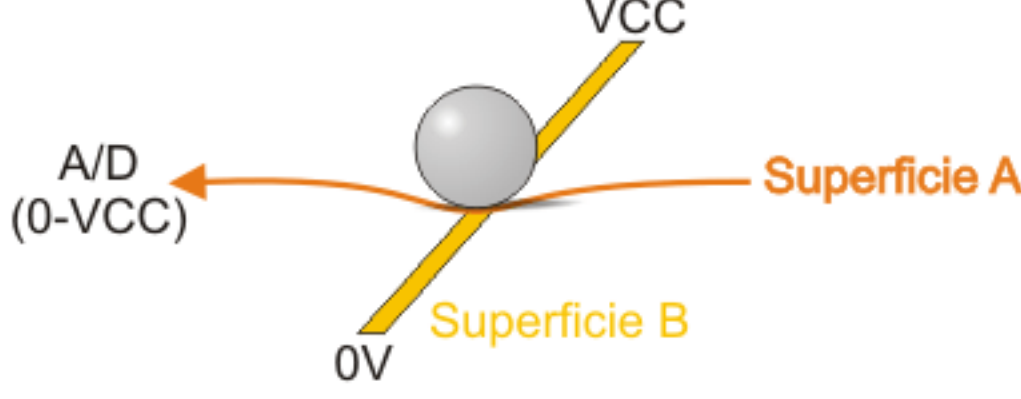
Un panel táctil es un panel fino, autoadhesivo y transparente, colocado sobre la pantalla de un LCD gráfico. Es muy sensible a la presión así que un suave toque provoca algunos cambios en la señal de salida. Hay diferentes tipos de paneles táctiles. El más sencillo es el panel táctil resistivo del que se hablará aquí.



Un panel táctil está compuesto por dos láminas rígidas, formando una estructura de ‘sándwich’ que tiene capas resistivas en sus caras internas. La resistencia de estas capas no excede normalmente de 1Kohm. Los lados opuestos de las láminas disponen de los contactos para acceder a un cable plano.



El procedimiento para determinar las coordenadas de la posición del panel que ha sido presionado se puede dividir en dos pasos. El primero es determinación de la coordenada X, y el segundo es de determinar la coordenada Y de la posición. Para determinar la coordenada X, es necesario conectar el contacto izquierdo en la superficie A a la masa (tierra) y el contacto derecho a la fuente de alimentación. Esto permite obtener un divisor de voltaje al presionar el panel táctil. El valor de voltaje obtenido en el divisor se puede leer en el contacto inferior de la superficie B. El voltaje variará en el rango de 0V al voltaje suministrado por la fuente de alimentación y depende de la coordenada X. Si el punto está próximo al contacto izquierdo de la superficie A, el voltaje estará próximo a 0V.



Para la determinación de la coordenada Y, es necesario conectar el contacto inferior de la superficie B a masa (tierra), mientras que el contacto superior se conectará a la fuente de alimentación. En este caso, el voltaje se puede leer en el contacto izquierdo de la superficie A. Para conectar un panel táctil al microcontrolador es necesario crear un circuito para el control del panel táctil. Por medio de este circuito, el microcontrolador conecta los contactos adecuados del panel táctil a masa y a la voltaje de alimentación (como describimos anteriormente) para determinar las coordenadas X y Y. El contacto inferior de la superficie B y el contacto izquierdo de la superficie A están conectados al convertidor A/D del microcontrolador. Las coordenadas X e Y se determinan midiendo el voltaje en los respectivos contactos. El software consiste en mostrar un menú en una pantalla LCD gráfica, conmutar de encendido a apagado del panel táctil (control del panel táctil) y leer los valores del convertidor A/D que representan realmente las coordenadas X e Y de la posición. Una vez determinadas las coordenadas, es posible decidir qué es lo que deseamos que haga el microcontrolador. En este ejemplo se explica cómo conmutar entre encendido y apagado dos pines digitales del microcontrolador, conectados a los LEDs A y B. En este ejemplo se utilizan las funciones que pertenecen a las librerías Glcd y ADC. Teniendo en cuenta que la superficie del panel táctil es ligeramente mayor que la del LCD gráfico, en caso de requerir una mayor precisión en la determinación de las coordenadas, es necesario incluir el software de calibración del panel táctil.

```
/* Cabecera *****/

// Conexiones del módulo Glcd
char GLCD_DataPort at PORTD;
sbit GLCD_CS1 at RB0_bit;
sbit GLCD_CS2 at RB1_bit;
sbit GLCD_RS at RB2_bit;
sbit GLCD_RW at RB3_bit;
sbit GLCD_EN at RB4_bit;
sbit GLCD_RST at RB5_bit;
sbit GLCD_CS1_Direction at TRISB0_bit;
sbit GLCD_CS2_Direction at TRISB1_bit;
sbit GLCD_RS_Direction at TRISB2_bit;
sbit GLCD_RW_Direction at TRISB3_bit;
sbit GLCD_EN_Direction at TRISB4_bit;
sbit GLCD_RST_Direction at TRISB5_bit;
// Final de conexiones del módulo Glcd

// Declaración de la cadena a visualizar en el GLCD
char msg1[] = "TOUCHPANEL EXAMPLE";
char msg2[] = "MIKROELEKTRONIKA";
char msg3[] = "BUTTON1";
char msg4[] = "BUTTON2";
char msg5[] = "RC6 OFF";
char msg6[] = "RC7 OFF";
char msg7[] = "RC6 ON ";
char msg8[] = "RC7 ON ";

// Declaración de variables globales
long x_coord, y_coord, x_coord128, y_coord64; // almacenar la posición de las
// coordenadas x e y

// Leer la coordenada X
unsigned int GetX() {
    //reading X
    PORTC.F0 = 1; // DRIVEA = 1 (electrodo izquierdo (LEFT) conectado, electrodo
    // derecho (RIGHT) conectado, electrodo superior (TOP)desconectado)
    PORTC.F1 = 0; // DRIVEB = 0 (electrodo inferior (BOTTOM) desconectado)
    Delay_ms(5);
    return ADC_Read(0); // leer el valor de X de RA0(BOTTOM)
}

// Leer la coordenada Y
unsigned int GetY() {
    //Leer la Y
    PORTC.F0 = 0; // DRIVEA = 0 (electrodo izquierdo (LEFT) desconectado, electrodo
    // derecho (RIGHT) desconectado, electrodo superior (TOP) conectado)
    PORTC.F1 = 1; // DRIVEB = 1 (electrodo inferior (BOTTOM) conectado)
    Delay_ms(5);
    return ADC_Read(1); // leer el valor de Y de RA1 (del eléctrodo izquierdo LEFT)
}

void main() {
    PORTA = 0x00;
    TRISA = 0x03; // RA0 y RA1 son entradas analógicas
    ANSEL = 0x03;
    ANSELH = 0; // Configurar otros pines AN como digitales de E/S
    PORTC = 0; // Todos los pines del puerto PORTC están a 0 (incluyendo los
    // pines RC6 y RC7)

    TRISC = 0; // PORTC es una salida

    // Inicialización del GLCD
    Glcd_Init(); // Glcd_Init_EP5
    Glcd_Set_Font(FontSystem5x7_v2, 5, 7, 32); // Seleccionar el tamaño de fuente 5x7
    Glcd_Fill(0); // Borrar GLCD
    Glcd_Write_Text(msg1,10,0,1);
    Glcd_Write_Text(msg2,17,7,1);

    // Visualizar botones en el GLCD:
    Glcd_Rectangle(8,16,60,48,1);
    Glcd_Rectangle(68,16,120,48,1);
    Glcd_Box(10,18,58,46,1);
    Glcd_Box(70,18,118,46,1);

    // Visualizar los mensajes en los botones
    Glcd_Write_Text(msg3,14,3,0);
    Glcd_Write_Text(msg5,14,4,0);
    Glcd_Write_Text(msg4,74,3,0);
    Glcd_Write_Text(msg6,74,4,0);

    while (1) {
        // leer X-Y y convertirlo en la resolución de 128x64 pixeles
        x_coord = GetX();
        y_coord = GetY();
        x_coord128 = (x_coord * 128) / 1024;
        y_coord64 = 64 - ((y_coord * 64) / 1024);

        //Si BUTTON1 ha sido presionado
        if ((x_coord128 >= 10) && (x_coord128 <= 58) && (y_coord64 >= 18) &&
            (y_coord64 <= 46)) {
            if(PORTC.F6 == 0) { // Si RC6 = 0
                PORTC.F6 = 1; // Invertir el estado lógico del pin RC6
                Glcd_Write_Text(msg7,14,4,0); // Visualizar un nuevo mensaje: RC6 ON
            }
            else { // Si RC6 = 1
                PORTC.F6 = 0; // Invertir el estado lógico del pin RC6
                Glcd_Write_Text(msg5,14,4,0); // Visualizar un nuevo mensaje: RC6 OFF
            }
        }

        // Si BUTTON2 ha sido presionado
        if ((x_coord128 >= 70) && (x_coord128 <= 118) && (y_coord64 >= 18) &&
            (y_coord64 <= 46)) {
            if(PORTC.F7 == 0) { // Si RC7 = 0
                PORTC.F7 = 1; // Invertir el estado lógico del pin RC7
                Glcd_Write_Text(msg8,74,4,0); // Visualizar un nuevo mensaje: RC7 ON
            }
            else { // Si RC7 = 1
                PORTC.F7 = 0; // Invertir el estado lógico del pin RC7
                Glcd_Write_Text(msg6,74,4,0); // Visualizar un nuevo mensaje: RC7 OFF
            }
        }
        Delay_ms(100);
    }
}
```

Para que este ejemplo funcione apropiadamente, es necesario marcar las siguientes librerías en la ventana Library Manager antes de compilar el programa.

- GLCD
- ADC
- C_Stdlib

SUBSCRIBE TO

- 📧 newsletter 🌐 github in linkedin f facebook 📺 youtube 📷 instagram 🐦 twitter

JOIN US

- Careers Make a Click Internship

COMPANY

- | | |
|--------------|----------|
| About us | Contact |
| Leadership | PressKit |
| Distributors | Timeline |
| Terms | |

TOOLCHAINS

- | | |
|-------|-------|
| PIC | dsPIC |
| PIC32 | ARM |
| AVR | FT90X |
| 8051 | PSOC |
| CEC | |

RESOURCES

- | | |
|-------------|------------|
| mikroBUS™ | mikroSDK |
| Click Cloud | Premium TS |
| Hexiwear™ | Libstock™ |
| eBooks | Outlet |
| Legacy | |