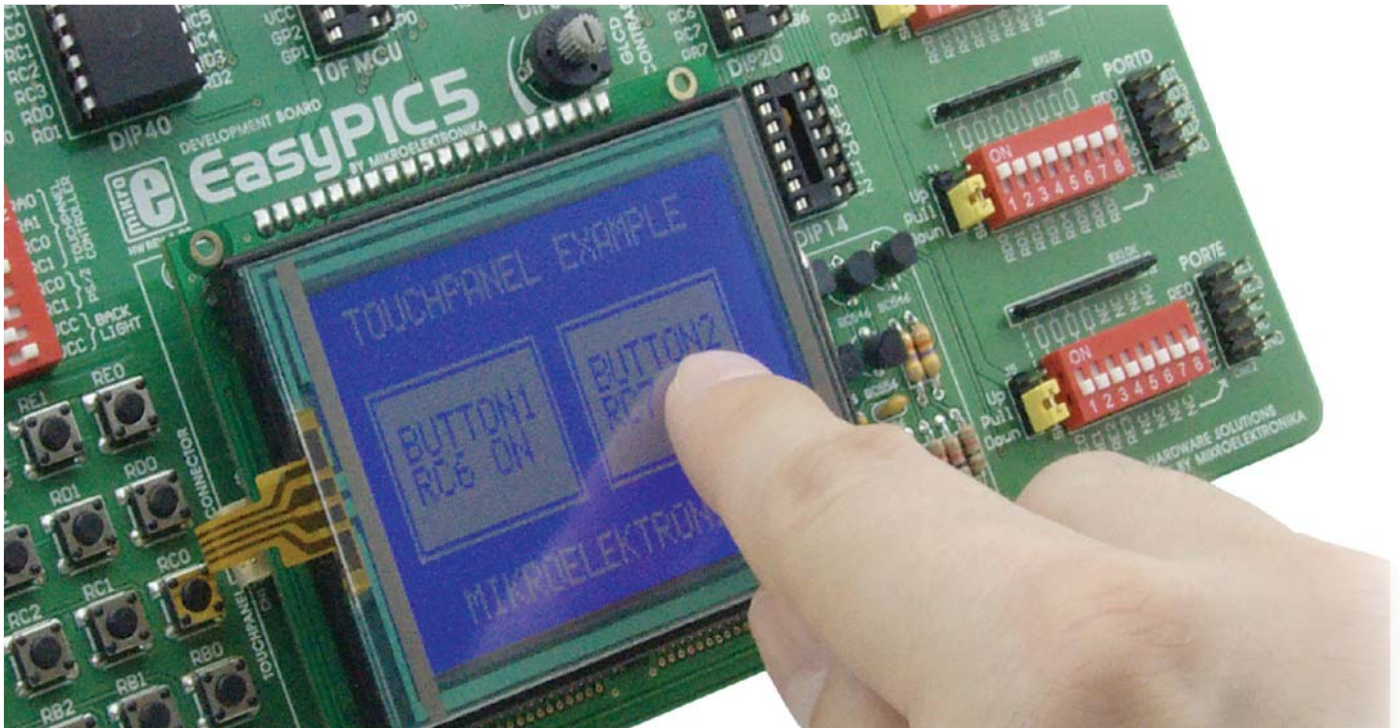


De acuerdo. Ahora necesitas una ... pantalla táctil



Por Dusan Mihajlovic

Departamento de Hardware de Mikroelektronika

¿Quieres que tu último proyecto tenga un interfaz sencillo e intuitivo? Si la respuesta es SI, una pantalla LCD gráfica con un panel táctil es la mejor elección porque juntos constituyen una pantalla táctil (Glcd+Panel táctil = Pantalla táctil). De esta manera, con un reducido número de componentes electrónicos serás capaz de crear un dispositivo atractivo y fácil de usar.

¿Qué es un panel táctil? Un panel táctil es un fino panel autoadhesivo colocado sobre la pantalla de un LCD gráfico. Es muy sensible a la presión de manera que un suave toque provoca algunos cambios en la señal de salida. Hay diferentes tipos de paneles táctiles. El más sencillo de ellos es el panel táctil resistivo que será del que hablaremos aquí.

Principio de funcionamiento

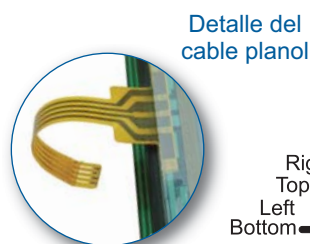
Un panel táctil resistivo está compuesto por dos láminas rígidas transparentes, formando una estructura "sándwich", que tienen una capa resistiva en sus caras internas. La resistencia de estas capas no excede normalmente de 1Kohm. Los lados opuestos de las láminas disponen de contactos para acceder a un cable plano. El procedimiento para determinar las coordenadas de la posición del panel que ha sido presionada puede dividirse en dos pasos. El primero es la determinación de la coordenada X y el segundo el de la coordenada Y del punto. Para determinar la coordenada X, es preciso conectar el contacto izquierdo de la superficie X a masa y el contacto derecho a la fuente de alimentación. Esto per-

mite obtener un divisor de tensión presionando el panel táctil. El valor de la tensión obtenida en el divisor se puede leer en el contacto inferior de la superficie Y. La tensión variará en el rango de 0 V a la tensión suministrada por la fuente de alimentación y depende de la coordenada X. Si el punto está próximo al contacto izquierdo de la superficie X, la tensión estará próxima a 0 V. Para la determinación de la coordenada Y, es preciso conectar el contacto inferior de la superficie Y

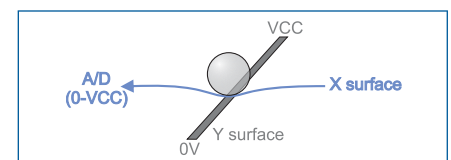
a masa, mientras que el contacto superior se conectará a la fuente de alimentación. En este caso, la lectora de la tensión se hará en el contacto izquierdo de la superficie X.

Conexión a un microcontrolador

Para conectar un panel táctil al microcontrolador, es preciso 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 tensión de alimentación (como



Detalle del cable plano



Determinación de la coordenada Y

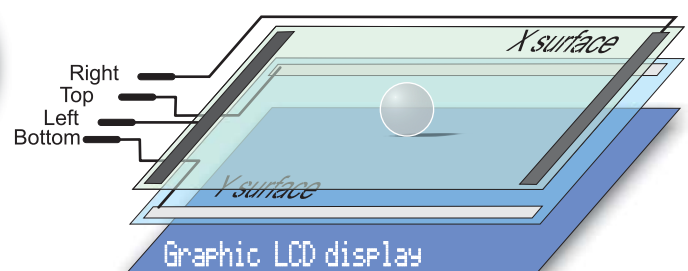
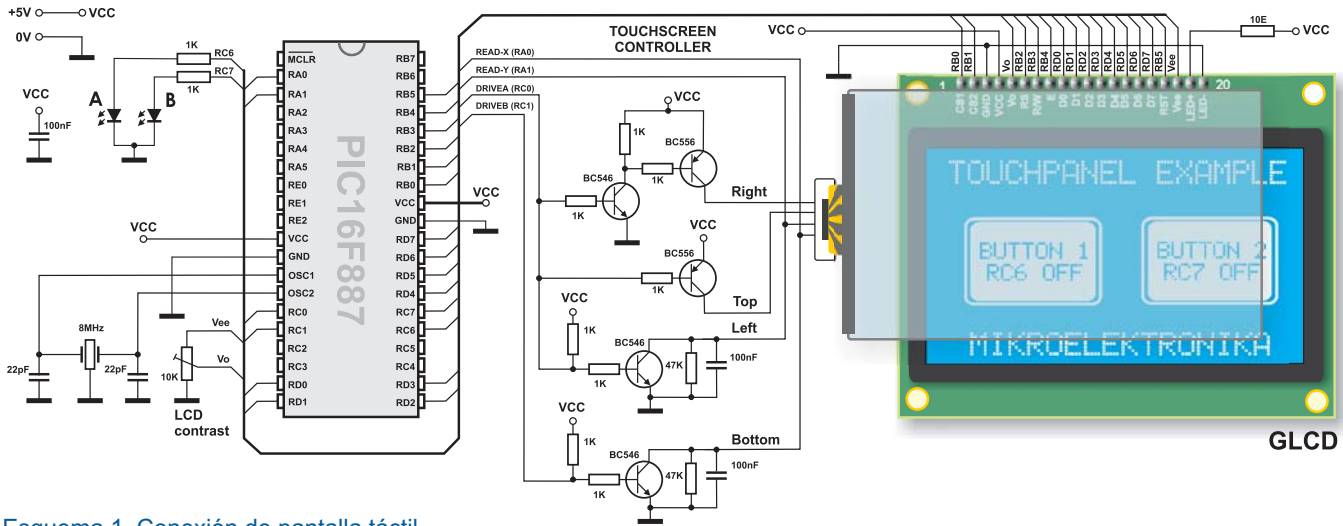


Figura 1. Estructura interna del panel táctil



Esquema 1. Conexión de pantalla táctil

describimos anteriormente) para determinar las coordenadas X e Y (véase el esquema 1). El contacto inferior de la superficie Y y el contacto izquierdo de la superficie X están conectados al convertidor A/D del microcontrolador. Las coordenadas X e Y se determinan midiendo la tensión en los respectivos contactos. En 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 al microcontrolador. Para ilustrarlo, podemos dar un vistazo al Ejemplo 1. En él se explica cómo conmutar entre "on" y "off" dos pines digitales del microcontrolador, conectados a los LEDs A y B, empleando para ello una pantalla LCD gráfica y un panel táctil.



Conector de cable plano en la placa antes ...



... y después de conectar el panel táctil.

Teniendo en cuenta que la superficie del panel táctil es ligeramente mayor que la del LCD gráfico, en el caso de querer mayor precisión en la determinación de las coordenadas, es preciso incluir el software de calibración del panel táctil.



Funciones utilizadas en el programa

- ADC_Read() Read analog value
- Delay_ms() Delay
- Glcd_box() Draw filled box*
- Glcd_circle() Draw circle
- Glcd_Dot() Draw dot
- Glcd_Fill() Delete/fill display*
- Glcd_H_Line() Draw horizontal line
- Glcd_Image() Import image
- Glcd_Init() LCD display initialization*
- Glcd_Line() Draw line
- Glcd_Read_Data() Read data from LCD
- Glcd_Rectangle() Draw rectangle*
- Glcd_Set_Font() Select font*
- Glcd_Set_Page() Select page
- Glcd_Set_Side() Select side of display
- Glcd_Set_X() Determine X coordinate
- Glcd_V_Line() Draw vertical line
- Glcd_Write_Char() Write character
- Glcd_Write_Data() Write data
- Glcd_Write_Text() Write text*

* Glcd library functions used in the program

Editor de bibliotecas mikroC for PIC® con bibliotecas listas para utilizar como: Ethernet, CAN, SD/MMC etc.

Ejemplo 1: Programa para demostrar el funcionamiento de la pantalla táctil

```

const char msg1[] = "TOUCHPANEL EXAMPLE";
const char msg2[] = "MIKROELEKTRONIKA";
const char msg3[] = "BUTTON1";
const char msg4[] = "BUTTON2";
const char msg5[] = "RC6 OFF";
const char msg6[] = "RC7 OFF";
const char msg7[] = "RC6 ON";
const char msg8[] = "RC7 ON";

long x_coord, y_coord, x_coord128, y_coord64;
char msg[16];

char * CopyConst2Ram(char * dest, const char * src)
{
    for(*dest++ = *src++);
    return dest;
}

unsigned int GetX() {
    //reading X
    PORTC.F0 = 1; //DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off)
    PORTC.F1 = 0; //DRIVEB = 0 (BOTTOM drive off)
    Delay_ms(5);
    return ADC_read(0); // reading X value from RA0 (BOTTOM)
}

unsigned int GetY() {
    //reading Y
    PORTC.F0 = 0; //DRIVEA = 0 (LEFT drive off, RIGHT drive off, TOP drive on)
    PORTC.F1 = 1; //DRIVEB = 1 (BOTTOM drive on)
    Delay_ms(5);
    return ADC_read(1); // reading Y value from RA1 (from LEFT)
}

void main() {
    PORTA = 0x00; // RA0 i RA1 are analog inputs
    TRISA = 0x03; // Configure other AN pins as digital I/O
    ANSEL = 0x03;
    ANSELH = 0;

    PORTC = 0; // PORTC is output
    TRISC = 0;

    Glcd_Init(&PORTB, 0, 1, 2, 3, 5, 4, &PORTD); // Glcd_Init_EP5
    Glcd_Set_Font(FontSystems5x8, 5, 8, 32); // Choose font
    Glcd_Fill(0); // Clear GLCD

    CopyConst2Ram(msg,msg1); // Copy "TOUCHPANEL EXAMPLE" string to RAM
    Glcd_Write_Text(msg,10,0,1);
    CopyConst2Ram(msg,msg2); // Copy "MIKROELEKTRONIKA" string to RAM
    Glcd_Write_Text(msg,17,7,1); //Display Buttons on 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);
    CopyConst2Ram(msg,msg3); // Copy "BUTTON1" string to RAM
    Glcd_Write_Text(msg,14,3,0);
    CopyConst2Ram(msg,msg5); // Copy "RC6 OFF" string to RAM
    Glcd_Write_Text(msg,14,4,0);
    CopyConst2Ram(msg,msg4); // Copy "BUTTON2" string to RAM
    Glcd_Write_Text(msg,74,3,0);
    CopyConst2Ram(msg,msg6); // Copy "RC7 OFF" string to RAM
    Glcd_Write_Text(msg,74,4,0);

    while (1) { // read X-Y and convert it to 128x64 space
        x_coord = GetX();
        y_coord = GetY();
        x_coord128 = (x_coord * 128) / 1024;
        y_coord64 = 64 - (y_coord * 64) / 1024;

        //if BUTTON1 is selected
        if ((x_coord128 >= 10) && (x_coord128 <= 58) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
            if(PORTC.F6 == 0) {
                PORTC.F6 = 1;
                CopyConst2Ram(msg,msg7); // Copy "RC6 ON" string to RAM
                Glcd_Write_Text(msg,14,4,0);
            }
            else {
                PORTC.F6 = 0;
                CopyConst2Ram(msg,msg5); // Copy "RC6 OFF" string to RAM
                Glcd_Write_Text(msg,14,4,0);
            }
        }

        //if BUTTON2 is selected
        if ((x_coord128 >= 70) && (x_coord128 <= 118) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
            if(PORTC.F7 == 0) {
                PORTC.F7 = 1;
                CopyConst2Ram(msg,msg8); // Copy "RC7 ON" string to RAM
                Glcd_Write_Text(msg,74,4,0);
            }
            else {
                PORTC.F7 = 0;
                CopyConst2Ram(msg,msg6); // Copy "RC7 OFF" string to RAM
                Glcd_Write_Text(msg,74,4,0);
            }
        }
        Delay_ms(100);
    }
}
    
```



NOTA: El código de este ejemplo para microcontroladores PIC® escrito en C, Basic y Pascal, así como los programas escritos para microcontroladores AVR® y dsPIC® se pueden encontrar en www.mikroe.com/en/articulo/