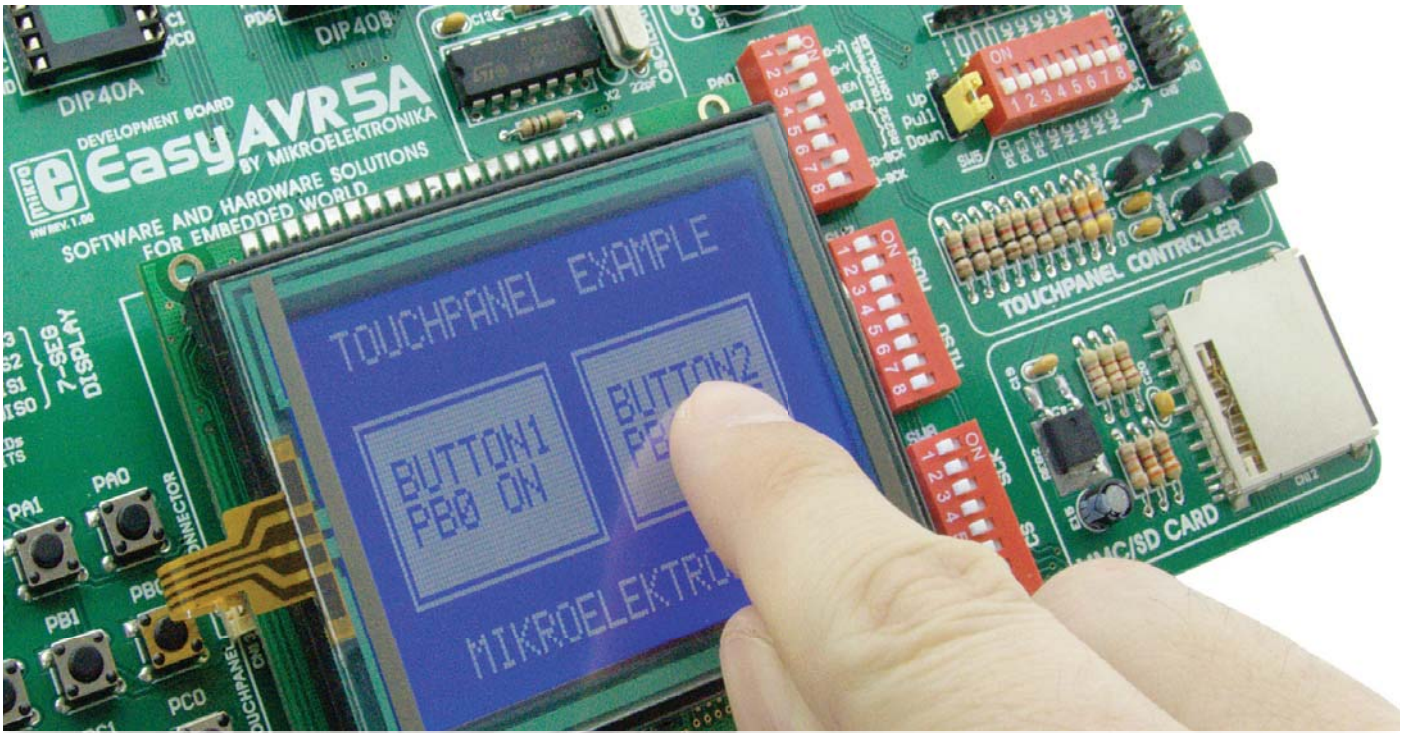


De acuerdo. Ahora necesitas una ... pantalla táctil



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

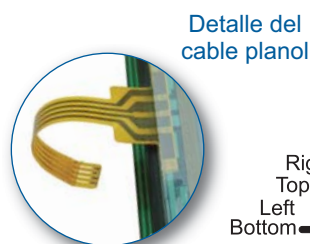
Un panel táctil resistivo está compuesto por dos láminas rígidas transparentes, formando una estructura "sándwich", que tienen una capa resistiva en sus caras internas. La resistencia de estas capas no excede normalmente de 1Kohm. Los lados opuestos de las láminas disponen de contactos para acceder a 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

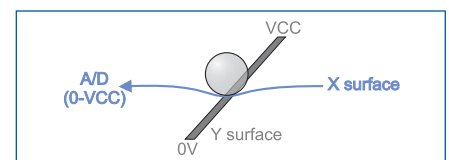
a masa, mientras que el contacto superior se conectará a la fuente de alimentación. En este caso, la lectora de la tensión se hará en el contacto izquierdo de la superficie X.

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

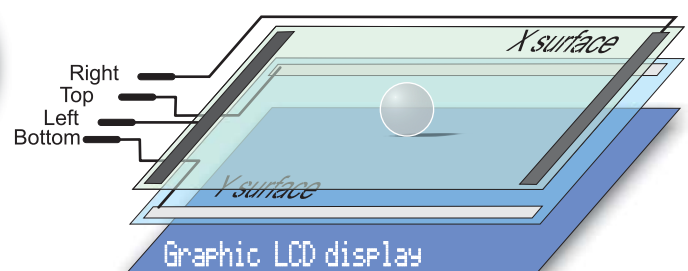
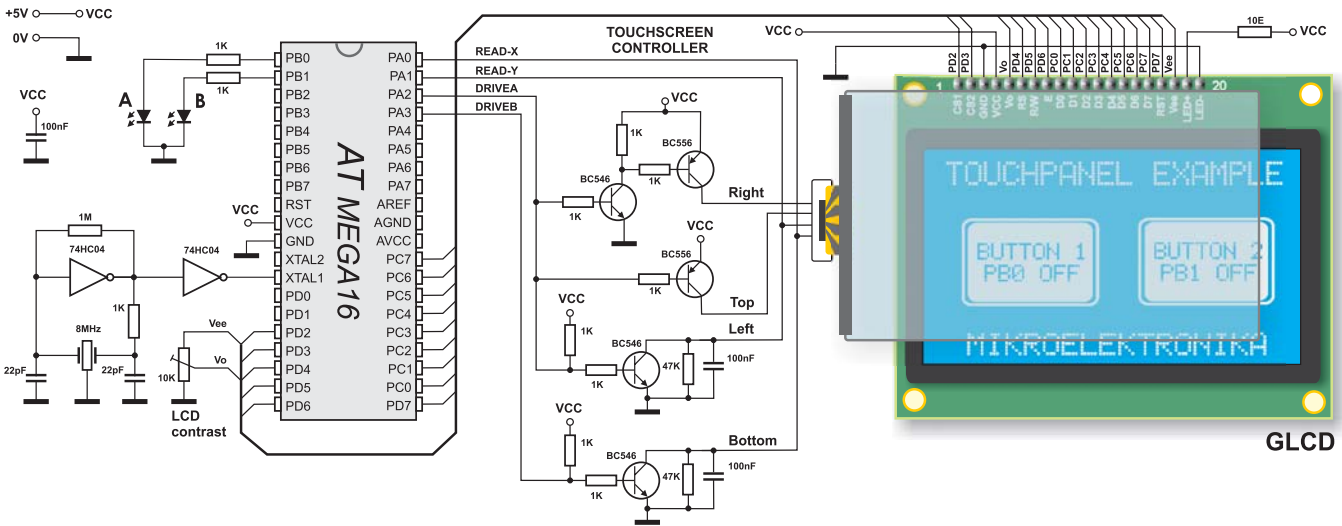


Figura 1. Estructura interna del panel táctil

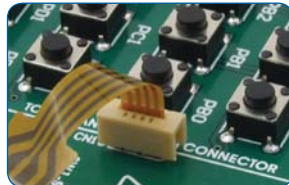


Esquema 1. Conexión de pantalla táctil

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 convertor 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 convertor 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.



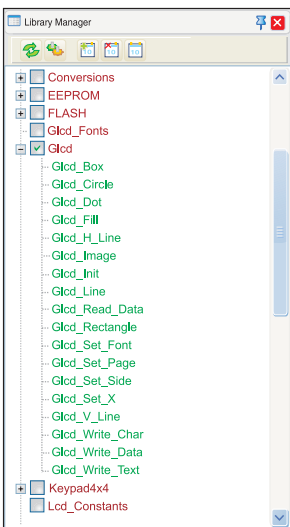
Conector de cable plano en la placa antes ...



... 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

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

```

program TouchPanel;
var GLCD_DataPort : byte at PORTC;    GLCD_DataPort_Direction : byte at DDRC; // Glcd module connections
var GLCD_CS1 : sbit at PORTD.B2;    GLCD_CS1_Direction : sbit at DDRD.B2;
var GLCD_CS2 : sbit at PORTD.B3;    GLCD_CS2_Direction : sbit at DDRD.B3;
var GLCD_RS : sbit at PORTD.B4;    GLCD_RS_Direction : sbit at DDRD.B4;
var GLCD_RW : sbit at PORTD.B5;    GLCD_RW_Direction : sbit at DDRD.B5;
var GLCD_EN : sbit at PORTD.B6;    GLCD_EN_Direction : sbit at DDRD.B6;
var GLCD_RST : sbit at PORTD.B7;    GLCD_RST_Direction : sbit at DDRD.B7; // End Glcd module connections

var DRIVE_A : sbit at PORTA.B2;    DRIVE_A_Direction : sbit at DDRA.B2;
var DRIVE_B : sbit at PORTA.B3;    DRIVE_B_Direction : sbit at DDRA.B3; // End Touch Panel module connections

var x_coord,y_coord,x_coord128,y_coord64 : longint; // scaled x-y position

function GetX() : word;
begin
    DRIVE_A := 1; //reading X
    DRIVE_B := 0; //DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off)
    Delay_ms(5); //DRIVEB = 0 (BOTTOM drive off)
    result := ADC_Read(0); // READ-X (BOTTOM)
end;

function GetY() : word;
begin
    DRIVE_A := 0; //reading Y
    DRIVE_B := 1; //DRIVEA = 0 (LEFT drive off, RIGHT drive off, TOP drive on)
    Delay_ms(5); //DRIVEB = 1 (BOTTOM drive on)
    result := ADC_Read(1); // READ-Y (LEFT)
end;

begin
    DRIVE_A_Direction := 1; // Set DRIVE_A pin as output
    DRIVE_B_Direction := 1; // Set DRIVE_B pin as output
    PORTB.B0 := 0; // Set PB0 pin as output (Default value 0)
    DDRB.B0 := 1; // Set PB0 pin as output (Default value 0)
    PORTB.B1 := 0; // Set PB1 pin as output (Default value 0)
    DDRB.B1 := 1; // Set PB1 pin as output (Default value 0)

    Glcd_Init(); // Initialize GLCD
    Glcd_Set_Font(@font5x7, 5, 7, 32); // Choose font, see __Lib_GLCDFonts.c in Uses folder
    Glcd_Fill(0); // Clear GLCD

    Glcd_Write_Text("TOUCHPANEL EXAMPLE;10,0,1);
    Glcd_Write_Text("MIKROELEKTRONIKA;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);
    Glcd_Write_Text("BUTTON1;14,3,0);
    Glcd_Write_Text("PBO OFF;14,4,0);
    Glcd_Write_Text("BUTTON2;74,3,0);
    Glcd_Write_Text("PB1 OFF;74,4,0);

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

        //if BUTTON1 is selected
        if ((x_coord128 >= 10) and (x_coord128 <= 58) and (y_coord64 >= 18) and (y_coord64 <= 46)) then
        begin
            if(PORTB.B0 = 0) then
            begin
                PORTB.B0 := 1;
                Glcd_Write_Text("PBO ON;14,4,0);
            end
            else
            begin
                PORTB.B0 := 0;
                Glcd_Write_Text("PBO OFF;14,4,0);
            end;
        end;

        //if BUTTON2 is selected
        if ((x_coord128 >= 70) and (x_coord128 <= 118) and (y_coord64 >= 18) and (y_coord64 <= 46)) then
        begin
            if(PORTB.B1 = 0) then
            begin
                PORTB.B1 := 1;
                Glcd_Write_Text("PB1 ON;74,4,0);
            end
            else
            begin
                PORTB.B1 := 0;
                Glcd_Write_Text("PB1 OFF;74,4,0);
            end;
        end;
        Delay_ms(100); // while true
    end;
end;
    
```

Editor de bibliotecas mikroPascal PRO for AVR® con bibliotecas listas para utilizar como: Ethernet, CAN, SD/MMC etc.

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

