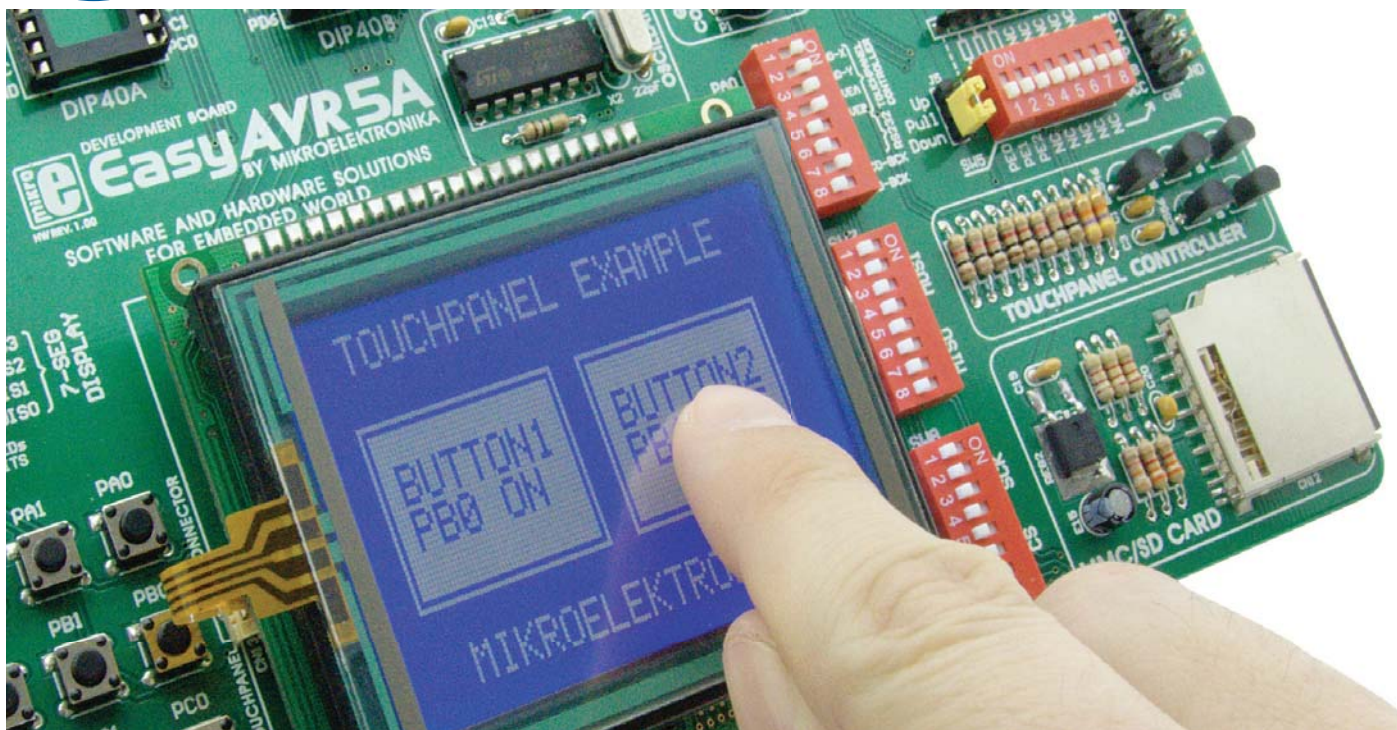


# OK. Vama je potreban... TOUCHSCREEN



Dušan Mihajlović

Mikroelektronika - odsek za hardver

Želite li da vaš novi uređaj ima jednostavan i intuitivan interfejs? Touchscreen u kombinaciji sa grafičkim displejom je idealna stvar za to. Sa malim brojem elektronskih komponenti napravićete veoma atraktivan uređaj koji će uz to biti i izuzetno jednostavan za korišćenje.

Šta je touch panel? Touch panel je transparentna staklena pločica koja je osetljiva na dodir i koja se uz pomoć samolepljivog sloja pričvršćuje za površinu ekrana grafičkog displeja. Postoji nekoliko vrsta touch panela od kojih je najjednostavniji rezistivni tip koji ćemo ovom prilikom prezentovati.

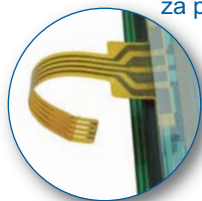
## Princip rada

Rezistivni touch panel se sastoji od dve providne pločice koje formiraju "sendvič" i koje su sa unutrašnje strane prevučene providnim slojem čija otpornost obično ne prelazi 1Kohm. Po dve naspramne stranice ovih pločica su izvedene u vidu kontakata koji su dostupni za upotrebu preko flet kabla.

Postupak određivanja koordinata tačke u kojoj je touch panel pritisnut odvija se u dva koraka. Prvi korak je određivanje X koordinata pritisnute tačke nakon čega se pristupa drugom koraku, odnosno određivanju Y koordinata pritisnute tačke. Da bismo odredili X koordinatu, neophodno je povezati kontakt na levoj strani X površine na masu i kontakt na desnoj strani X površine na napon napajanja. Ovim se omogućuje da se pritiskom na površinu touch panela dobije razdelnik napona čija se vrednost zatim čita sa donjeg kontakta Y površine. Vrednost napona će biti u

granicama od 0V do vrednosti napona napajanja i menjaće se u zavisnosti od vrednosti X koordinate. Ukoliko je touch panel pritisnut bliže levom kontaktu X površine vrednost očitano napona će biti bliže vrednosti od 0V. Da bismo odredili Y koordinatu pritisnute tačke neophodno je povezati kontakt na donjoj strani Y površine na masu, a kontakt na gornjoj strani Y površine na napon napajanja. U ovom slučaju, vrednost napona se čita sa levog kontakta X površine.

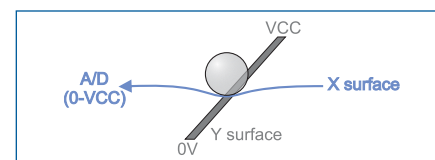
Trakasti kabl za povezivanje



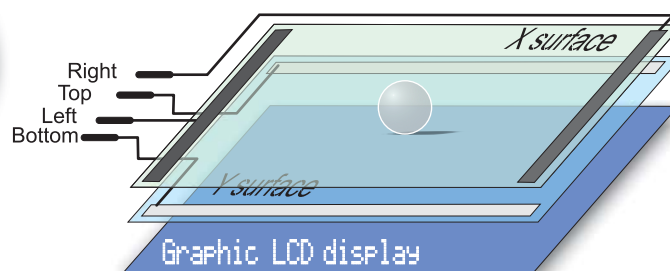
## Povezivanje sa mikrokontrolerom

Da bismo povezali touch panel sa mikrokontrolerom, neophodno je napraviti električno kolo za kontrolu touch panela. Pomoću ovog kola mikrokontroler povezuje odgovarajuće kontakte touch panela na masu i napon napajanja (kao što je opisano) u cilju određivanja X i Y koordinata (Šema 1).

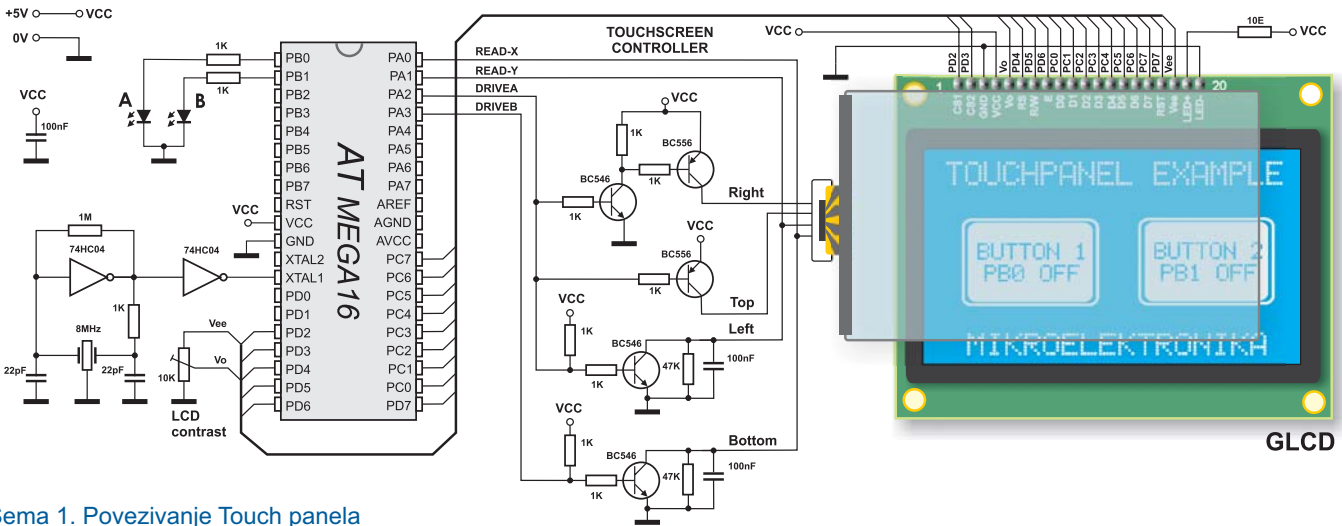
Kontakt na donjoj strani Y površine, odnosno kontakt na levoj strani X površine povezani su na A/D konvertor mikrokontrolera. Merenjem na-



Određivanje Y koordinate



Slika 1. Unutrašnja struktura touch panela



Šema 1. Povezivanje Touch panela

pona na ovim kontaktima određuje se X, odnosno Y koordinata pritisnute tačke. Softver koji je neophodan da bi upravljao ovim uređajem je veoma jednostavan. Suština njegovog rada je ispisivanje menija na grafičkom displeju, uključenje/isključenje električnog kola za kontrolu touch panela i čitanje vrednosti A/D konvertora koje zapravo predstavljaju X i Y koordinate pritisnute tačke.

Kada odredimo vrednosti koordinata, na osnovu njih možemo odrediti šta želimo da mikrokontroler uradi. Kao prost primer (Primer 1) prikazano je kako se uz pomoć displeja i touch panela može izvršiti uključenje/isključenje dva digitalna pina na mikrokontroleru (LED diode A i B).



Konektor za povezivanje trakastim kablom pre...



...i posle priključenja touch panela.

S obzirom da je površina touch panela malo veća od površine grafičkog displeja, u slučaju da želite veću preciznost određivanja koordinata pritisnute tačke, neophodno je uraditi softversku kalibraciju touch panela.

Primer 1: Program za demonstraciju rada touch panela

```

program TouchPanel;
var GLCD_DataPort : byte at PORTC;    GLCD_DataPort_Direction : byte at DDRC;
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;

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;

var x_coord,y_coord,x_coord128,y_coord64 : longint;

function GetX() : word;
begin
    DRIVE_A := 1;
    DRIVE_B := 0;
    Delay_ms(5);
    result := ADC_Read(0);
end;

function GetY() : word;
begin
    DRIVE_A := 0;
    DRIVE_B := 1;
    Delay_ms(5);
    result := ADC_Read(1); // READ-X (LEFT)
end;

begin
    DRIVE_A_Direction := 1;
    DRIVE_B_Direction := 1;
    PORTB.B0 := 0;
    DDRB.B0 := 1;
    PORTB.B1 := 0;
    DDRB.B1 := 1;

    Glcd_Init();
    Glcd_Set_Font(@font5x7, 5, 7, 32);
    Glcd_Fill(0);

    Glcd_Write_Text("TOUCHPANEL EXAMPLE;10,0,1);
    Glcd_Write_Text("MIKROELEKTRONIKA;17,7,1);

    Glcd_Rectangle(8,16,60,48,1);
    Glcd_Rectangle(68,16,120,48,1);
    Glcd_Box(10,18,38,46,1);
    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();
        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;
                Gcd_Write_Text("PB1 OFF;74,4,0);
            end;
        end;
        Delay_ms(100);
    end;
end;
    
```

Spisak funkcija korišćenih u programu

ADC_Read()	Čitanje analogne vrednosti
Delay_ms()	Kašnjenje
Glcd_box()	Crtanje ispunjenog pravougaonika*
Glcd_circle()	Crtanje kružnice
Glcd_Dot()	Crtanje tačke
Glcd_Fill()	Brisanje/popuna ekrana*
Glcd_H_Line()	Crtanje horizontalne linije
Glcd_Image()	Unošenje slike
Glcd_Init()	Inicijalizacija LCD displeja*
Glcd_Line()	Crtanje linije
Glcd_Read_Data()	Čitanje podataka iz LCD-a
Glcd_Rectangle()	Crtanje pravougaonika*
Glcd_Set_Font()	Izbor fonta*
Glcd_Set_Page()	Izbor strane
Glcd_Set_Side()	Izbor polovine ekrana
Glcd_Set_X()	Određivanje X koordinate
Glcd_V_Line()	Crtanje vertikalne linije
Glcd_Write_Char()	Ispisivanje karaktera
Glcd_Write_Data()	Ispisivanje bilo kog podatka
Glcd_Write_Text()	Ispisivanje teksta*

\* Funkcije koje su iskoristene u programu.

**NOTE:** Ovaj program, pisan za AVR® mikrokontrolere u programima C, Basic i Pascal kao i programe napisane za mikrokontrolere PIC® i dsPIC® možete pronaći na našem web sajtu: [www.mikroe.com/en/article/](http://www.mikroe.com/en/article/)

