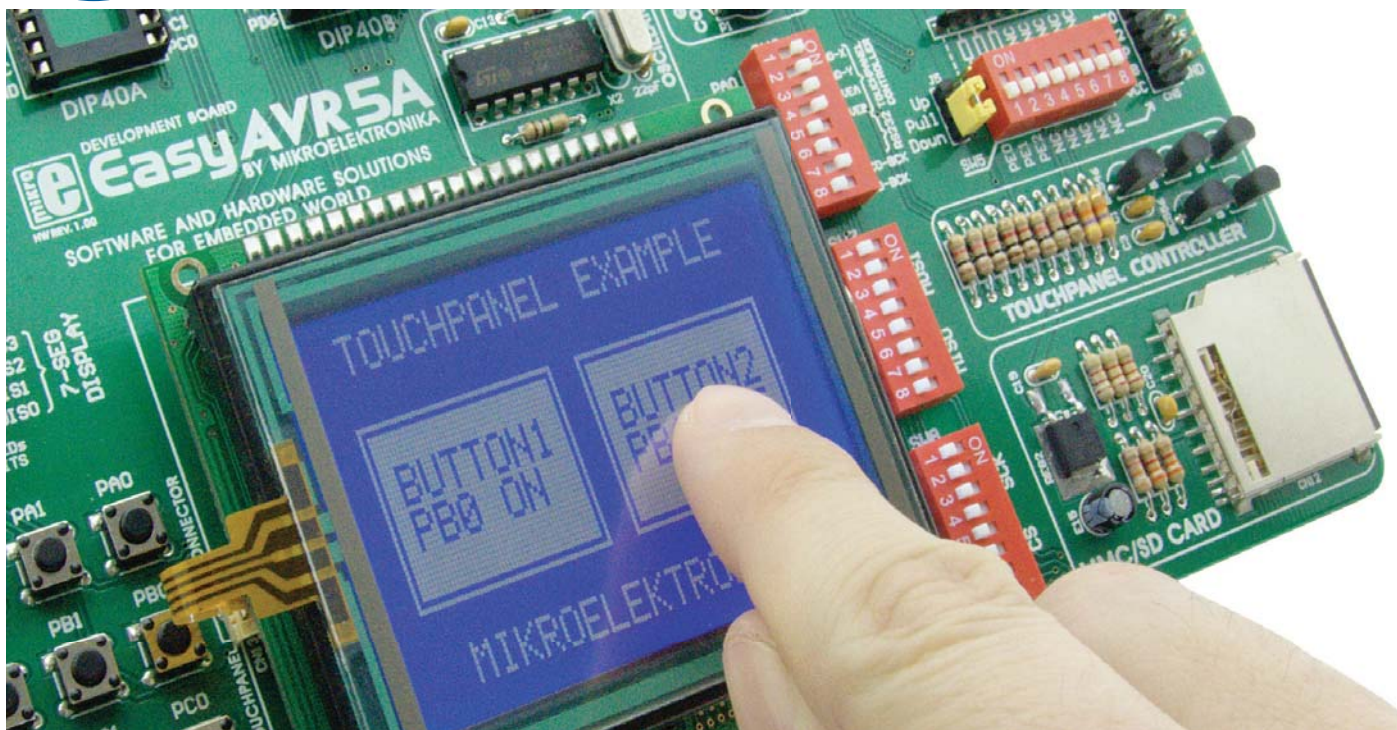


# OK. Vama je potreban... TOUCHSCREEN



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.

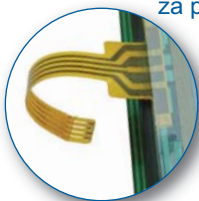
## Princip rada

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.

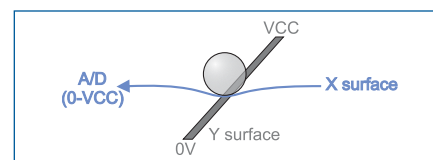
Trakasti kabl  
za povezivanje



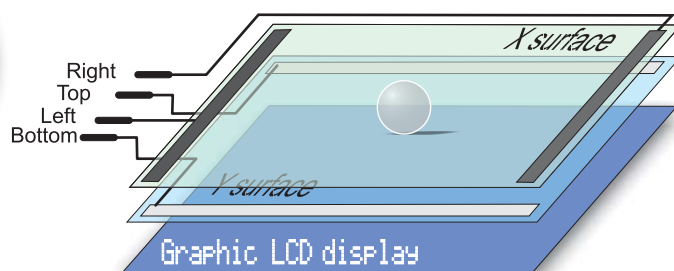
## Povezivanje sa mikrokontrolerom

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).

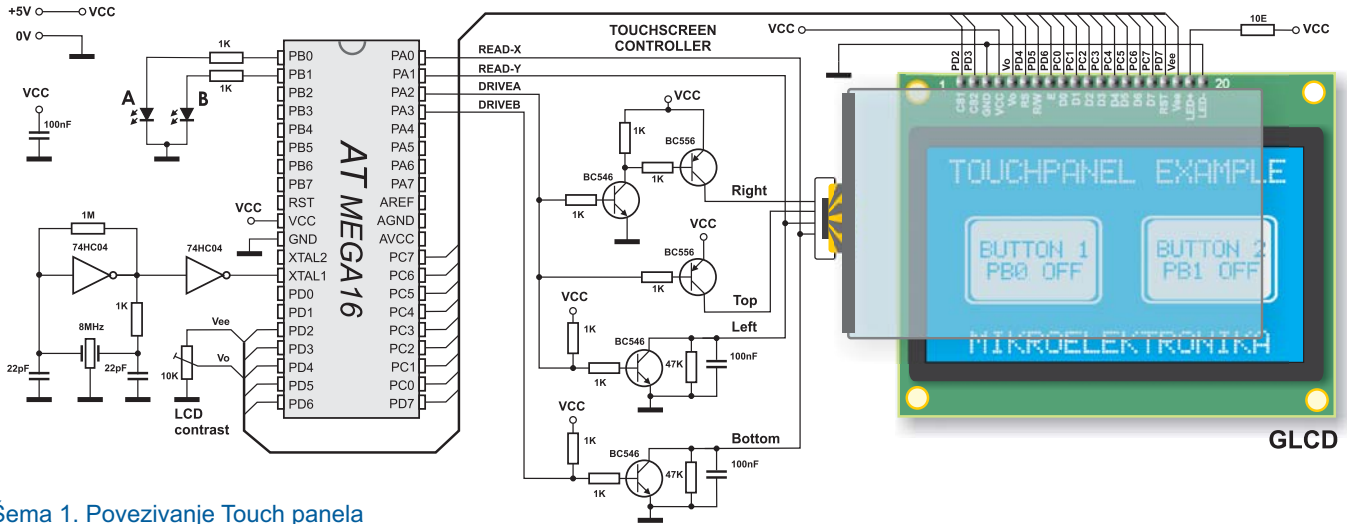
Kontakt na donjoj strani Y površine, odnosno kontakt na levoj strani X površine povezani su na A/D konvertor mikrokontrolera. Merenjem na-



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).



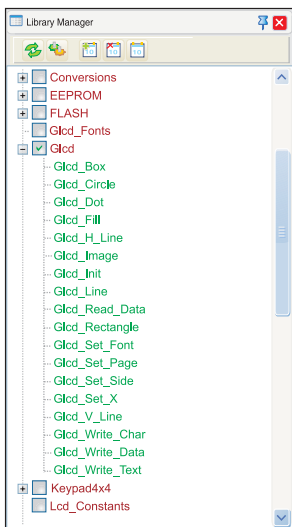
Konektor za povezivanje trakastim kablom pre...



...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.

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 iskoristišene u programu.

Primer 1: Program za demonstraciju rada touch panela

```
// Glcd module connections
char GLCD_DataPort at PORTC;

sbit GLCD_CS1 at PORTD.B2;
sbit GLCD_CS2 at PORTD.B3;
sbit GLCD_RS at PORTD.B4;
sbit GLCD_RW at PORTD.B5;
sbit GLCD_EN at PORTD.B6;
sbit GLCD_RST at PORTD.B7;

char GLCD_DataPort_Direction at DDRD;
sbit GLCD_CS1_Direction at DDRD.B2;
sbit GLCD_CS2_Direction at DDRD.B3;
sbit GLCD_RS_Direction at DDRD.B4;
sbit GLCD_RW_Direction at DDRD.B5;
sbit GLCD_EN_Direction at DDRD.B6;
sbit GLCD_RST_Direction at DDRD.B7;
// End Glcd module connections

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

long x_coord, y_coord, x_coord128, y_coord64; // scaled x-y position

unsigned int GetX() { //reading X
    DRIVE_A = 1; // DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off)
    DRIVE_B = 0; // DRIVEB = 0 (BOTTOM drive off)
    Delay_ms(5);
    return ADC_Read(0); // READ-X (BOTTOM)
}

unsigned int GetY() { //reading Y
    DRIVE_A = 0; // DRIVEA = 0 (LEFT drive off, RIGHT drive off, TOP drive on)
    DRIVE_B = 1; // DRIVEB = 1 (BOTTOM drive on)
    Delay_ms(5);
    return ADC_Read(1); // READ-X (LEFT)
}

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

    Glcd_Init(); // Initialize GLCD
    Glcd_Fill(0); // Clear GLCD
    Glcd_Set_Font(font5x7, 5, 7, 32); // Choose font,
    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); //Display Buttons on GLCD:
    Glcd_Box(10,18,58,46,1);
    Glcd_Box(70,18,118,46,1);
    Glcd_Write_Text("BUTTON1",14,3,0);
    Glcd_Write_Text("PB0 OFF",14,4,0);
    Glcd_Write_Text("BUTTON2",74,3,0);
    Glcd_Write_Text("PB1 OFF",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(PORTB.B0 == 0) {
                PORTB.B0 = 1;
                Glcd_Write_Text("PB0 ON",14,4,0);
            }
            else {
                PORTB.B0 = 0;
                Glcd_Write_Text("PB0 OFF",14,4,0);
            }
        }

        //if BUTTON2 is selected
        if ((x_coord128 >= 70) && (x_coord128 <= 118) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
            if(PORTB.B1 == 0) {
                PORTB.B1 = 1;
                Glcd_Write_Text("PB1 ON",74,4,0);
            }
            else {
                PORTB.B1 = 0;
                Glcd_Write_Text("PB1 OFF",74,4,0);
            }
        }

        Delay_ms(100);
    }
}
```



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/)