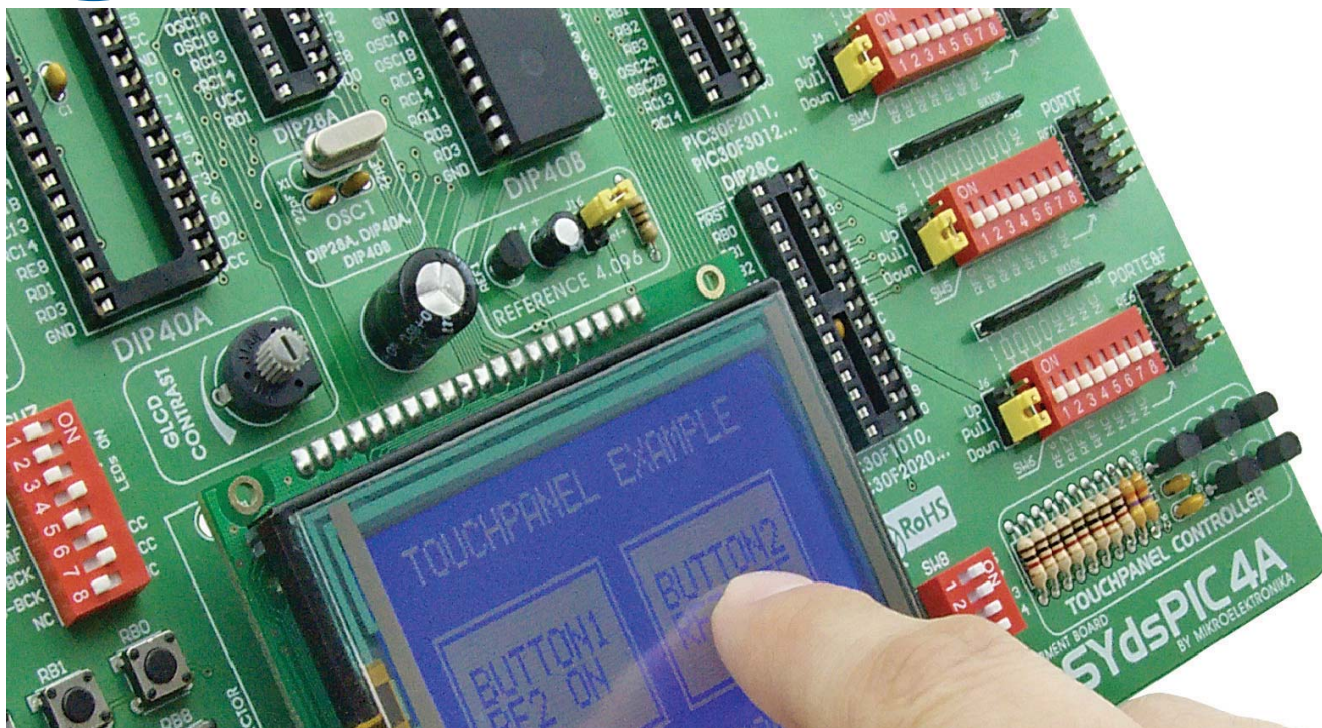


OK. Vama je potreban... TOUCHSCREEN



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.

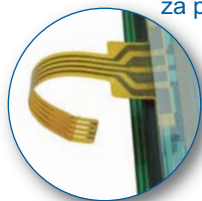
Princip rada

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 koordinate 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.

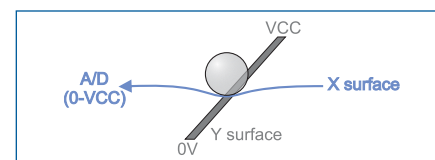
Trakasti kabl
za povezivanje



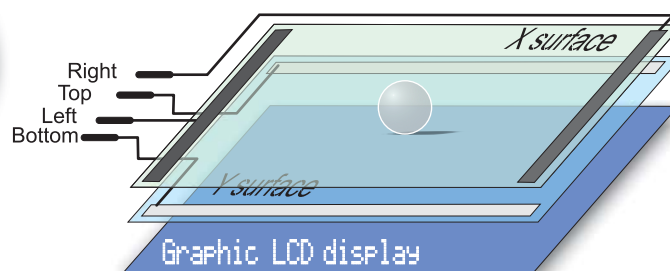
Povezivanje sa mikrokontrolerom

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).

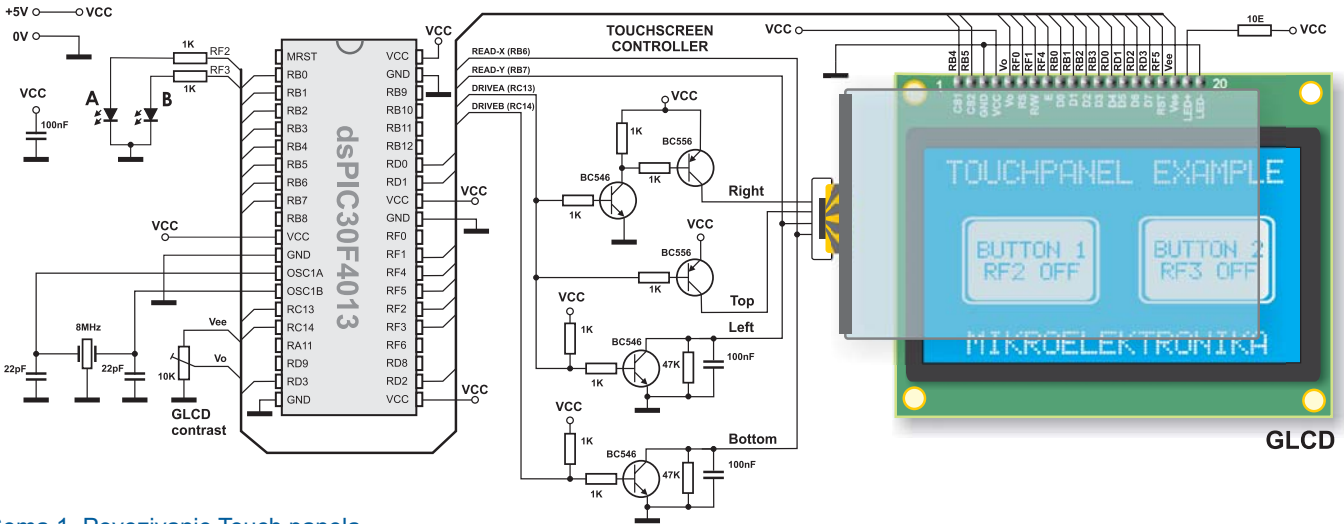
Kontakt na donjoj strani Y površine, odnosno kontakt na levoj strani X površine povezani su na A/D konvertor mikrokontrolera. Merenjem na-



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, uključivanje/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 uključivanje/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

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

unsigned int GetX() {
    // reading X
    LATC.F13 = 1; // DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off)
    LATC.F14 = 0; // DRIBEB = 0 (BOTTOM drive off)
    Delay_ms(5);
    return ADC_read(6); // reading X value from RB6 (BOTTOM)
}

unsigned int GetY() {
    // reading Y
    LATC.F13 = 0; // DRIVEA = 0 (LEFT drive off, RIGHT drive off, TOP drive on)
    LATC.F14 = 1; // DRIBEB = 1 (BOTTOM drive on)
    Delay_ms(5);
    return ADC_read(7); // reading Y value from RB7 (LEFT)
}

void main() {
    ADPCFG = 0xFF3F; // RB6 and RB7 are Analog Inputs

    LATC.F2 = 0; // RF2 is Output
    TRISCF2 = 0; // RF2 is Output
    LATC.F3 = 0; // RF2 is Output
    TRISCF3 = 0;

    LATC = 0; // RC13 is Output
    TRISC.F13 = 0; // RC14 is Output
    TRISC.F14 = 0;

    Glcd_Init_EasyDsPIC40();
    Glcd_Set_Font(FontSystem5x8, 5, 8, 32); // Choose font
    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("RF2 OFF",14,4,0);
    Glcd_Write_Text("BUTTON2",74,3,0);
    Glcd_Write_Text("RF3 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) / 4096;
        y_coord64 = 64 - (y_coord * 64) / 4096;

        // if BUTTON1 is selected
        if ((x_coord128 >= 10) && (x_coord128 <= 58) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
            if (PORTC.F2 == 0) {
                LATCF2 = 1;
                Glcd_Write_Text("RF2 ON",14,4,0);
            }
            else {
                LATCF2 = 0;
                Glcd_Write_Text("RF2 OFF",14,4,0);
            }
        }

        // if BUTTON2 is selected
        if ((x_coord128 >= 70) && (x_coord128 <= 118) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
            if (PORTC.F3 == 0) {
                LATCF3 = 1;
                Glcd_Write_Text("RF3 ON",74,4,0);
            }
            else {
                LATCF3 = 0;
                Glcd_Write_Text("RF3 OFF",74,4,0);
            }
        }
    }
    Delay_ms(100);
}
```

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 iskorištene u programu.



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

Pisano u kompaniji
mikroc
for dsPIC