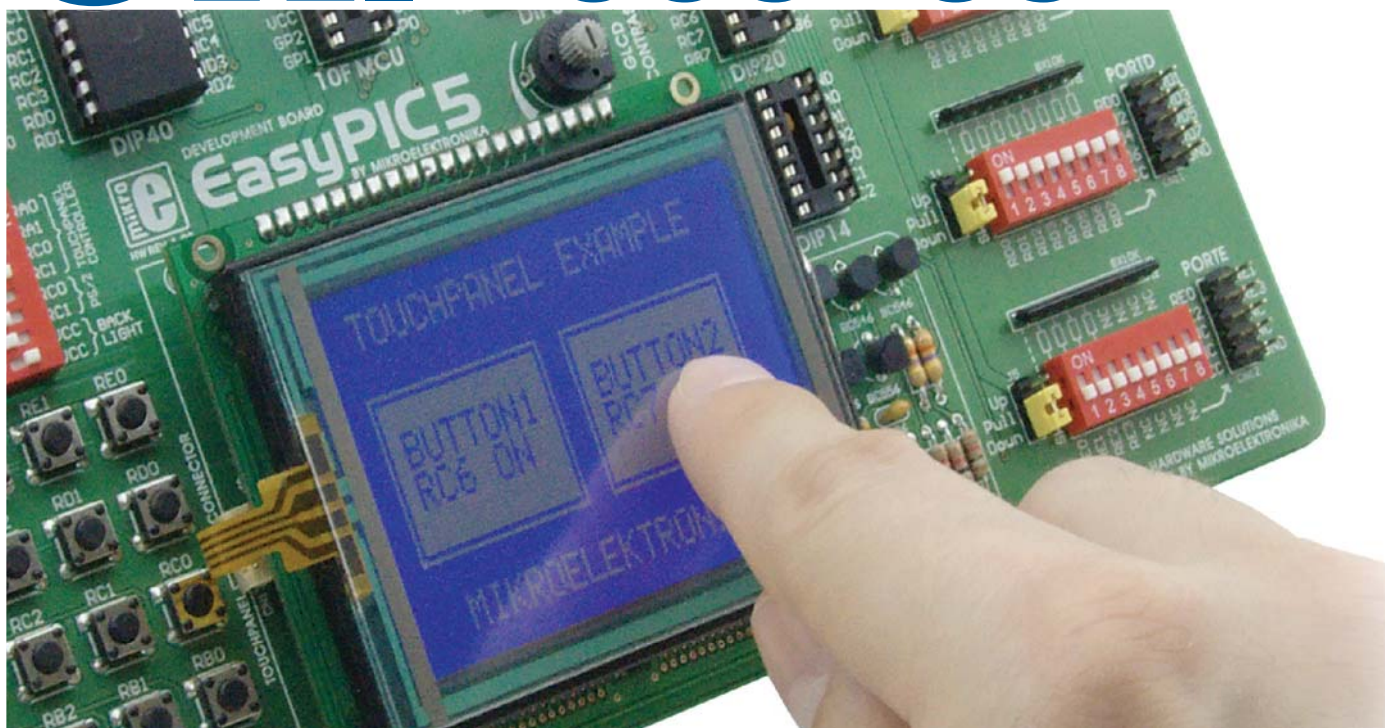


OK. Vama je potreban... TOUCHSCREEN



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.

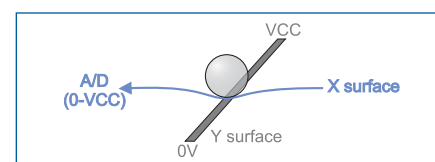
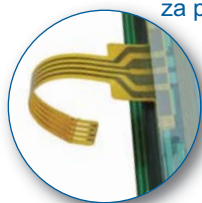
Princip rada

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.

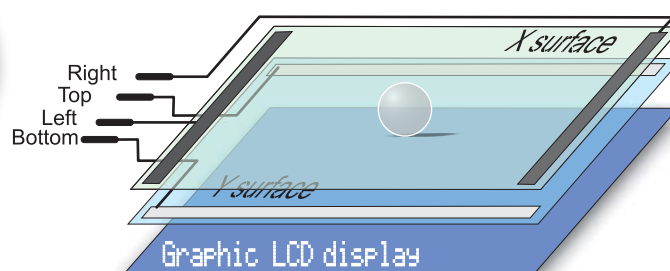
Postupak određivanja koordinata tačke u kojoj je touch panel pritisnut odvija se u dva koraka. Prvi korak je određivanje X koordinata pritisnute tačke nakon čega se pristupa drugom koraku, odnosno određivanju Y koordinata pritisnute tačke. Da bismo odredili X koordinatu, neophodno je povezati kontakt 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.

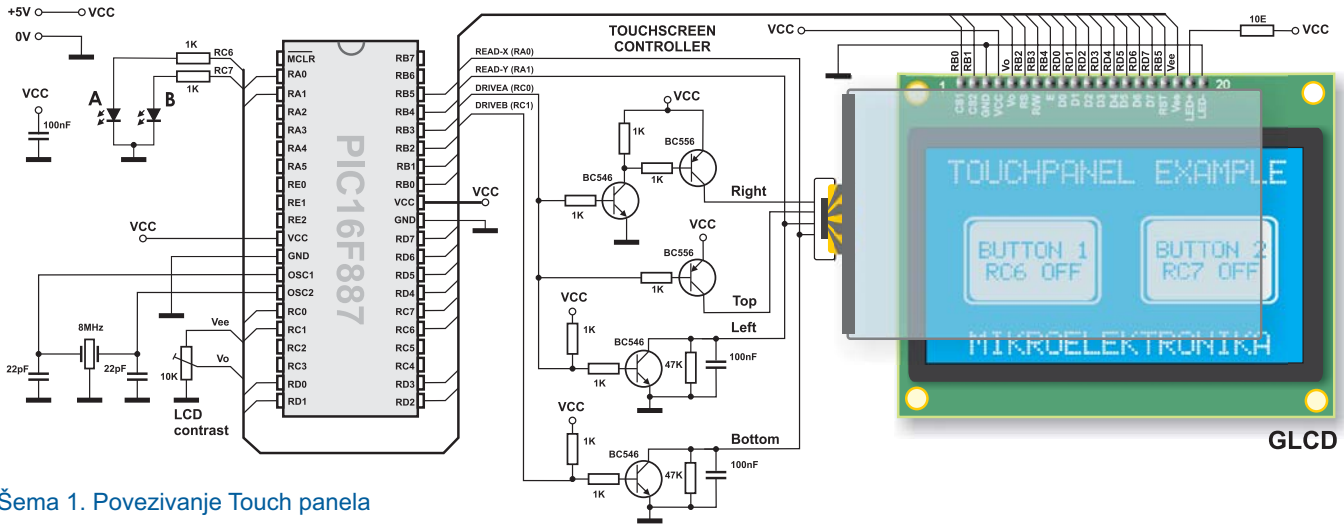
Trakasti kabl
za povezivanje



Određivanje Y koordinate



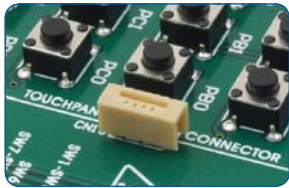
Slika 1. Unutrašnja struktura touch panela



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).

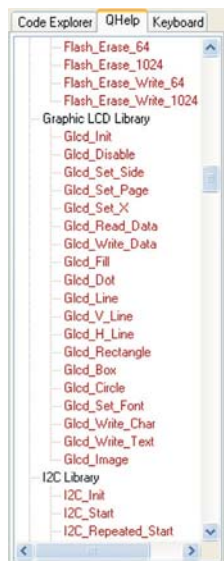


Konektor za povezivanje trakastim kablom pre...



...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 iskorištene u programu.

Primer 1: Program za demonstraciju rada touch panela

```

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

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

void main() {
PORTA = 0x00;
TRISA = 0x03; // RA0 i RA1 are analog inputs
ANSEL = 0x03; // Configure other AN pins as digital I/O
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); // Copy "MIKROELEKTRONIKA" string to RAM
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); // Copy "RC6 OFF" string to RAM
CopyConst2Ram(msg,msg5); // Copy "BUTTON2" string to RAM
Glcd_Write_Text(msg,14,4,0); // Copy "RC7 OFF" string to RAM
CopyConst2Ram(msg,msg4); // Copy "RC7 OFF" string to RAM
Glcd_Write_Text(msg,74,3,0); // Copy "RC7 OFF" string to RAM
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; // Copy "RC6 ON" string to RAM
CopyConst2Ram(msg,msg7); // Copy "RC6 ON" string to RAM
Glcd_Write_Text(msg,14,4,0);
}
else {
PORTC.F6 = 0; // Copy "RC6 OFF" string to RAM
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; // Copy "RC7 ON" string to RAM
CopyConst2Ram(msg,msg8); // Copy "RC7 ON" string to RAM
Glcd_Write_Text(msg,74,4,0);
}
else {
PORTC.F7 = 0; // Copy "RC7 OFF" string to RAM
CopyConst2Ram(msg,msg6); // Copy "RC7 OFF" string to RAM
Glcd_Write_Text(msg,74,4,0);
}
}
Delay_ms(100);
}
}
    
```

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/

