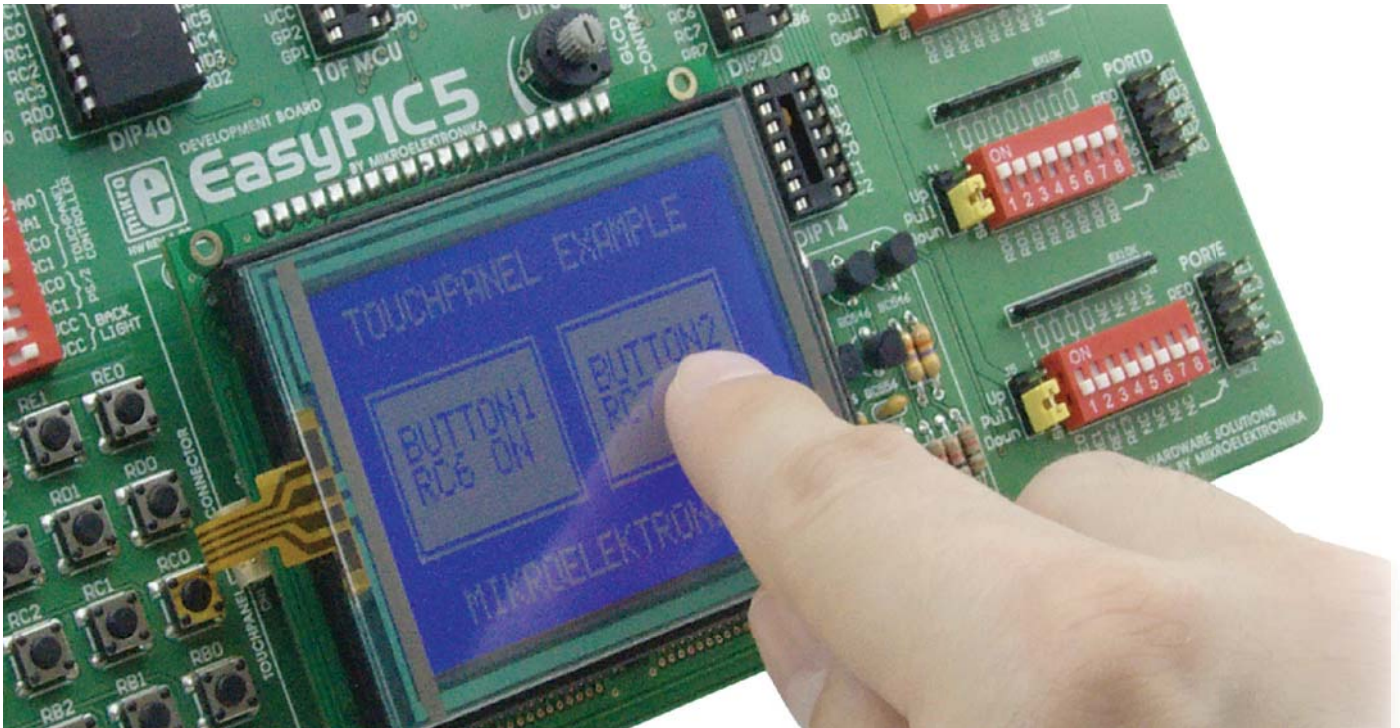


Oké. U wilt dus een... AANRAAKSCHEM



By Dusan Mihajlovic

Afdeling Hardware, Mikroelektronika

Wilt u voor uw nieuwste project een eenvoudige en intuïtieve interface? Als het antwoord JA is, is een grafisch lcd-scherm met aanraakpaneel de beste keuze, want samen vormen deze een aanraakscherm (Glcd + aanraakpaneel = aanraakscherm). Op die manier kunt u met een klein aantal elektronische componenten een aantrekkelijk en gebruiksvriendelijk apparaat maken.

Wat is een aanraakscherm eigenlijk? Een aanraakscherm is een dun, zelfklevend, doorzichtig paneel dat over een grafisch lcd-scherm wordt gelegd. Het is uiterst drukgevoelig, zodat zelfs een zachte aanraking al een verandering in het uitgangssignaal veroorzaakt. Er zijn verschillende soorten aanraakpanelen. Het eenvoudigste is het resistieve aanraakpaneel dat we hier zullen behandelen.

Werkingsprincipe

Een resistief aanraakpaneel bestaat uit twee doorzichtige harde folies die samen een "sandwichstructuur" vormen met resistieve lagen aan de binnenzijden. De weerstand van deze lagen is doorgaans niet hoger dan 1 Kohm. De tegenoverliggende zijden van de folies hebben contacten voor aansluiting aan een flatcable. Het proces waarmee de coördinaten worden bepaald van de plaats waar het aanraakpaneel wordt ingedrukt, kan in twee stappen worden onderverdeeld. De eerste stap bestaat uit het bepalen van de X-coördinaat, de tweede uit het bepalen van de Y-coördinaat van het betreffende punt. Om de X-coördinaat te kunnen bepalen is het noodzakelijk om het linker contact van het X-oppervlak te verbinden met massa en het rech-

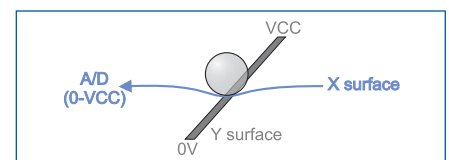
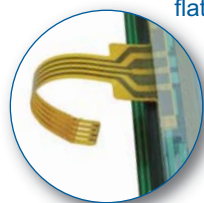
ter contact met de voedingsspanning. Hierdoor ontstaat een spanningsdeler wanneer op het aanraakpaneel wordt gedrukt. De waarde van de spanningsdeler wordt gemeten op het onderste contact van het Y-oppervlak. De spanning ligt ergens tussen 0 V en de voedingsspanning; dit is afhankelijk van de X-coördinaat. Als het punt dichterbij het linker contact van het X-oppervlak ligt, zal de spanning dichterbij de buurt van 0 V liggen. Om de Y-coördinaat te kunnen bepalen is het noodzakelijk het onderste contact van het Y-oppervlak (bottom) te verbinden met massa en

het bovenste contact met de voedingsspanning. In dit geval wordt de spanning afgelezen op het linkercontact van het X-oppervlak.

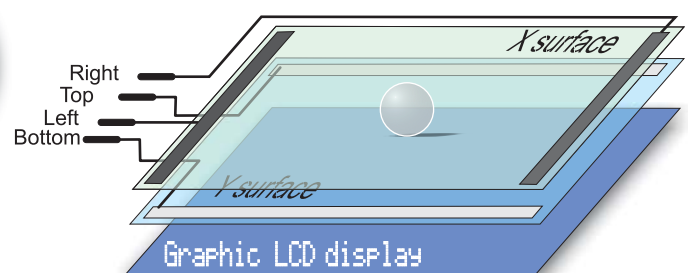
Aansluiting op een microcontroller

Om een aanraakpaneel te kunnen aansluiten op een microcontroller moet een circuit voor aanraakpaneelbesturing worden gemaakt. Met behulp van dit circuit kan de microcontroller de juiste contacten van het aanraakpaneel verbinden met massa en de voedingsspanning (zoals hierboven beschreven) om zo de X- en Y-coördi-

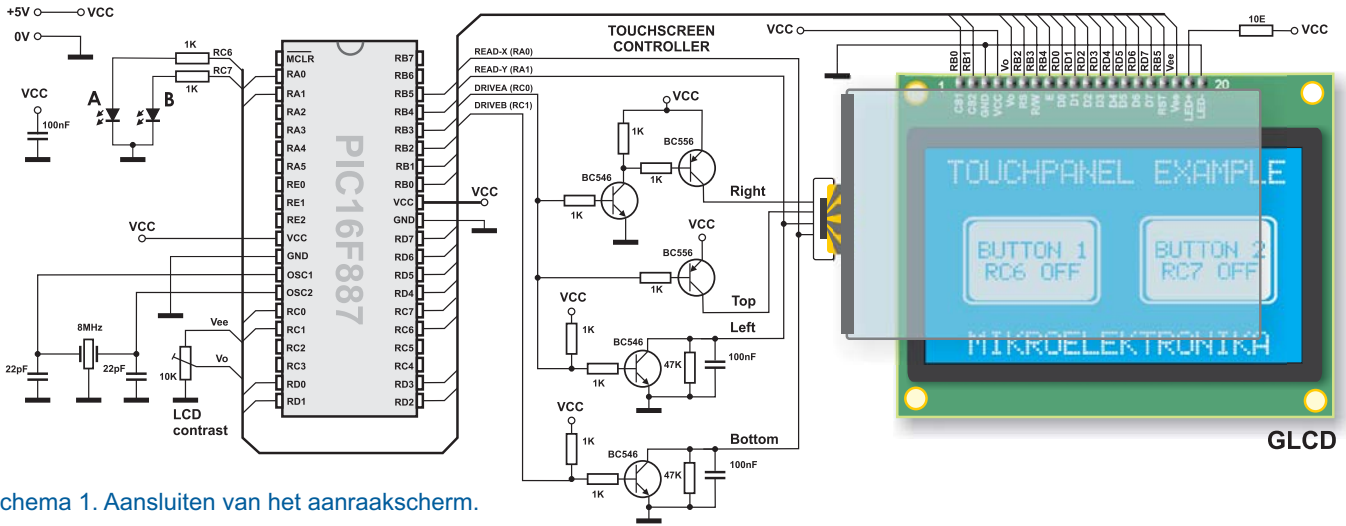
Detail flatcable



Bepaling van de Y-coördinaat



Figuur 1. Interne structuur van het aanraakpaneel



Schema 1. Aansluiten van het aanraakscherm.

naten te bepalen (zie schema 1). Het onderste contact van het Y-oppervlak en het linker contact van het X-oppervlak worden aangesloten op de A/D-omzetter van de microcontroller. De X- en Y-coördinaten worden bepaald door de spanning op de respectievelijke contacten te meten. De software bestaat uit het schrijven van een menu voor een grafisch lcd-scherm, waarbij het circuit voor aanraakpaneelbesturing in/uit wordt geschakeld en de waarden van de A/D-omzetter worden uitgelezen (deze vertegenwoordigen de X- en Y-coördinaten van het aanraakpunt). Zodra de coördinaten zijn bepaald, is het mogelijk te beslissen wat de microcontroller verder moet gaan doen. Kijk ter illustratie naar voorbeeld 1. Hierin is te zien hoe twee microcontroller-pennen worden gebruikt om LED A en B in en uit te schakelen met behulp van het display en het aanraakpaneel.

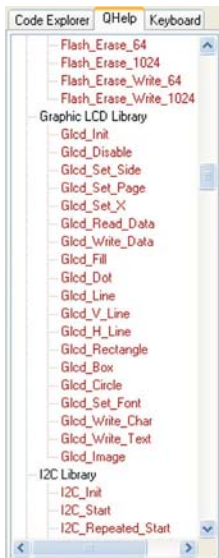


Kaartaansluiting voor flatcable vóór...



...en na het aansluiten van het aanraakpaneel.

Aangezien het oppervlak van het aanraakpaneel iets groter is dan het oppervlak van het grafische lcd-scherm, is het noodzakelijk om een software-kalibratie van het aanraakpaneel uit te voeren als u een grotere nauwkeurigheid wilt bereiken.



Functies die worden gebruikt in het programma

- ADC_Read() Lees analoge waarde
- Delay_ms() Wachtijid
- Glcd_box() Tekens ingevuld vak*
- Glcd_circle() Tekens cirkel
- Glcd_Dot() Tekens punt
- Glcd_Fill() Wis/vul display*
- Glcd_H_Line() Tekens horizontale lijn
- Glcd_Image() Importeer beeld
- Glcd_Init() Initialisatie lcd-scherm*
- Glcd_Line() Tekens lijn
- Glcd_Read_Data() Lees gegevens van lcd
- Glcd_Rectangle() Tekens rechthoek*
- Glcd_Set_Font() Selecteer font*
- Glcd_Set_Page() Selecteer pagina
- Glcd_Set_Side() Selecteer zijde van display
- Glcd_Set_X() Bepaal X-coördinaat
- Glcd_V_Line() Tekens verticale lijn
- Glcd_Write_Char() Schrijf tekens
- Glcd_Write_Data() Schrijf gegevens
- Glcd_Write_Text() Schrijf tekst*

* In het programma gebruikte Glcd-bibliotheekfuncties

Voorbeeld 1. Demoprogramma voor gebruik van een aanraakscherm

```

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; // RA0 i RA1 are analog inputs
    TRISA = 0x03;
    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);
    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);
    CopyConst2Ram(msg, msg5); // Copy "RC6 OFF" string to RAM
    Glcd_Write_Text(msg, 14, 4, 0);
    CopyConst2Ram(msg, msg4); // Copy "BUTTON2" string to RAM
    Glcd_Write_Text(msg, 74, 3, 0);
    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;
                CopyConst2Ram(msg, msg7); // Copy "RC6 ON" string to RAM
                Glcd_Write_Text(msg, 14, 4, 0);
            }
            else {
                PORTC.F6 = 0;
                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;
                CopyConst2Ram(msg, msg8); // Copy "RC7 ON" string to RAM
                Glcd_Write_Text(msg, 74, 4, 0);
            }
            else {
                PORTC.F7 = 0;
                CopyConst2Ram(msg, msg6); // Copy "RC7 OFF" string to RAM
                Glcd_Write_Text(msg, 74, 4, 0);
            }
        }
        Delay_ms(100);
    }
}
    
```



Opn.: De code voor dit voorbeeld voor PIC-microcontrollers in C, Basic en Pascal plus de programma's voor AVR's en dsPIC's zijn beschikbaar op onze website www.mikroe.com/en/article/