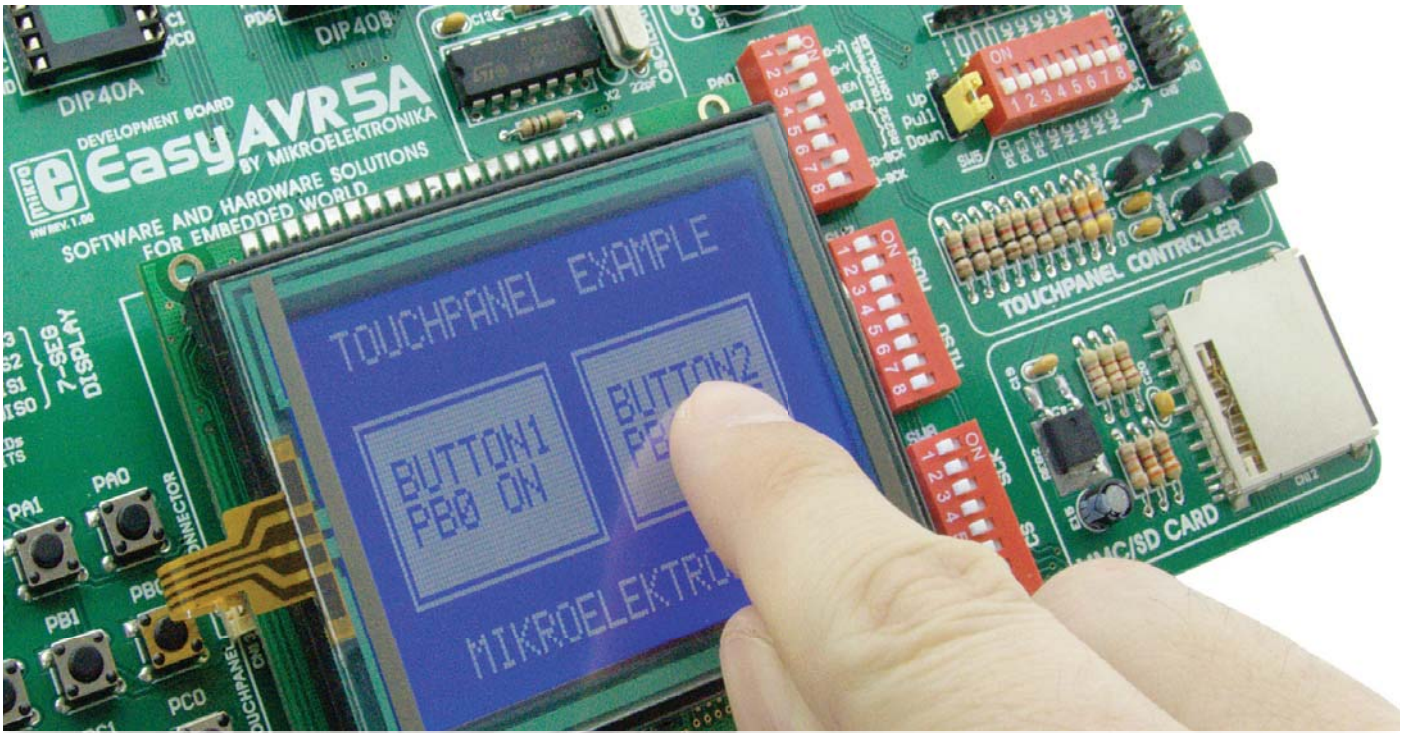


Oké. U wilt dus een... AANRAAKSCHERM



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-

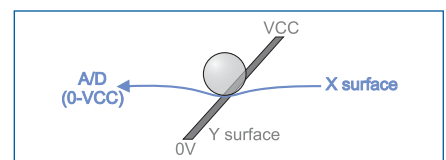
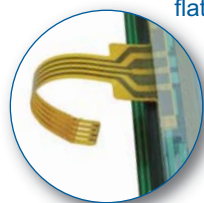
ter contact met de voedingsspanning. Hierdoor ontstaat een spanningsdeler wanneer op het aanraakpaneel wordt gedrukt. De waarde van de spanningsdeler wordt gemeten op het onderste contact van het Y-oppervlak. De spanning ligt ergens tussen 0 V en de voedingsspanning; dit is afhankelijk van de X-coördinaat. 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.

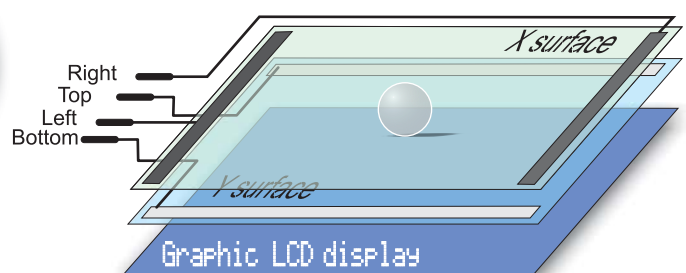
Aansluiting op een microcontroller

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-

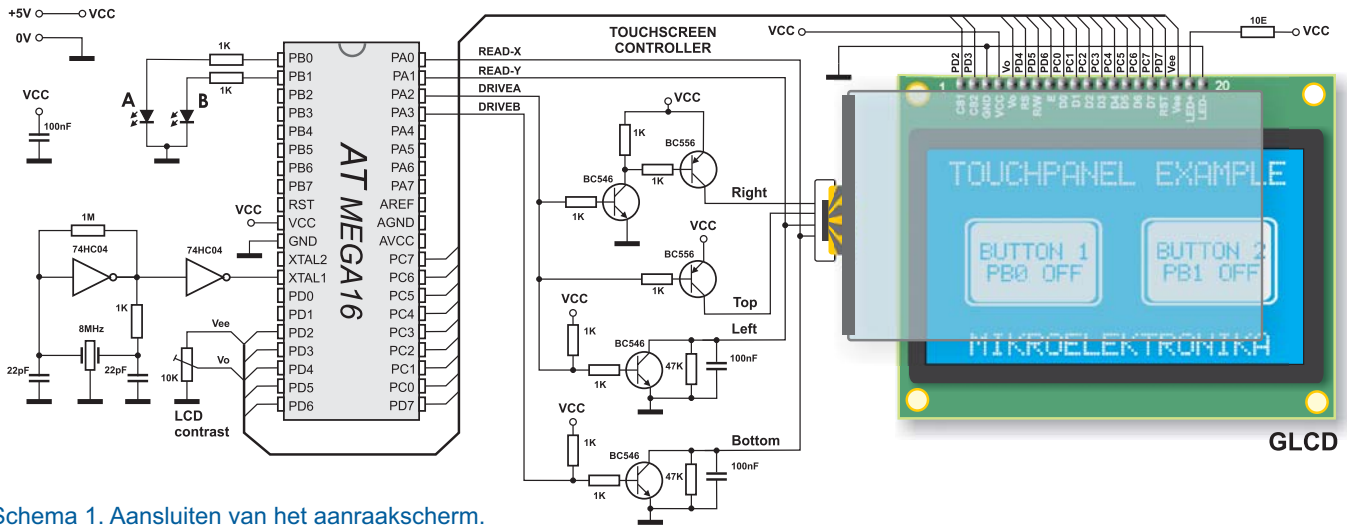
Detail flatcable



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

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

* In het programma gebruikte Glcd-bibliotheekfuncties

mikroPascal PRO for AVR®-bibliotheekeditor met kant-en-klare bibliotheken, zoals ethernet, CAN, SD/MMC, enz.

Voorbeeld 1. Demoprogramma voor gebruik van een aanraakscherm

```

program TouchPanel;
var GLCD_DataPort : byte at PORTC;    GLCD_DataPort_Direction : byte at DDRC; // Glcd module connections
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; // End Glcd module connections

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; // End Touch Panel module connections

var x_coord, y_coord, x_coord128, y_coord64 : longint; // scaled x-y position

function GetX() : word;
begin
  DRIVE_A := 1; // reading X
  DRIVE_B := 0; // DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off)
  Delay_ms(5); // DRIVEB = 0 (BOTTOM drive off)
  result := ADC_Read(0); // READ-X (BOTTOM)
end;

function GetY() : word;
begin
  DRIVE_A := 0; // reading Y
  DRIVE_B := 1; // DRIVEA = 0 (LEFT drive off, RIGHT drive off, TOP drive on)
  Delay_ms(5); // DRIVEB = 1 (BOTTOM drive on)
  result := ADC_Read(1); // READ-Y (LEFT)
end;

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

  Glcd_Init(0); // Initialize GLCD
  Glcd_Set_Font(@font5x7, 5, 7, 32); // Choose font, see __Lib_GlCDFonts.c in Uses folder
  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('PBO ON',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(); // read X-Y and convert it to 128x64 space
    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;
        Glcd_Write_Text('PB1 OFF',74,4,0);
      end;
    end;
  end;
  Delay_ms(100); // while true
end;

```



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