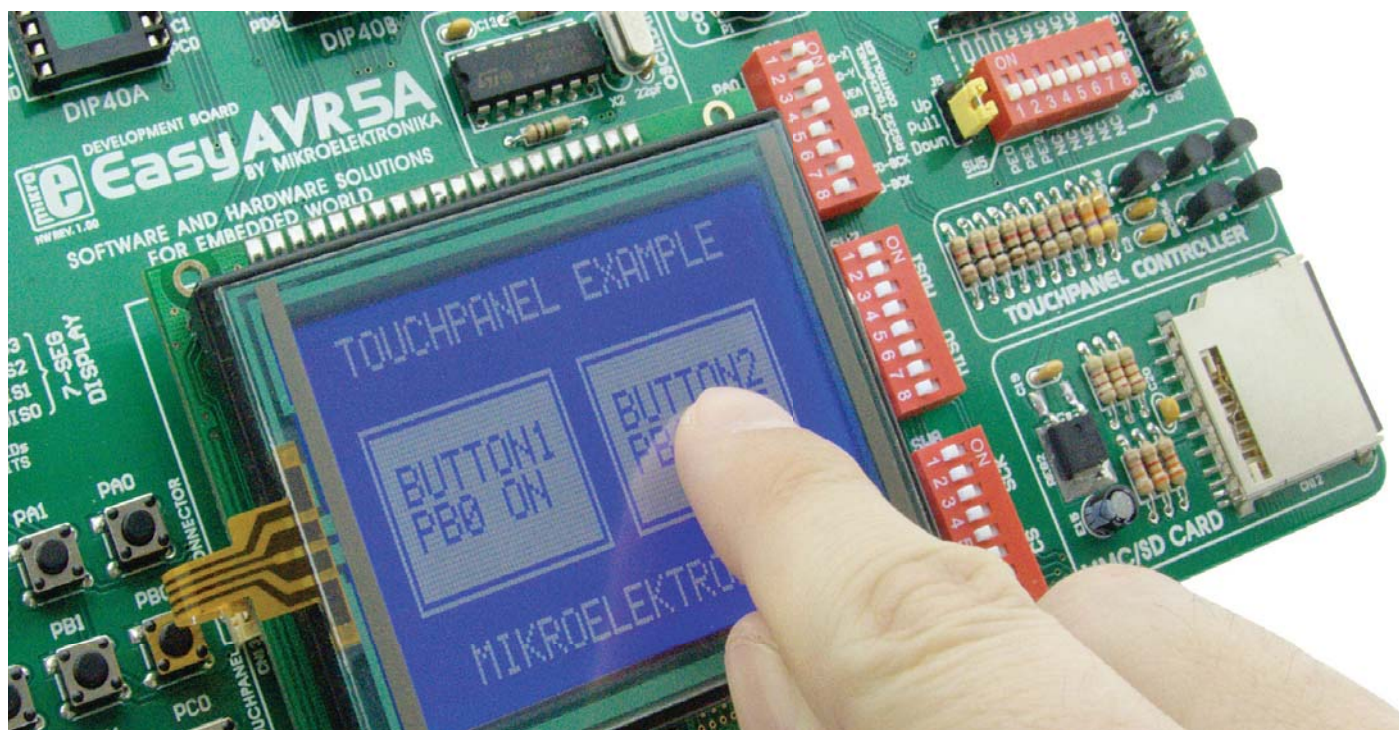


# Bon. Maintenant il vous faut un... écran tactile



Par Dusan Mihajlovic

mikroElektronika Département Matériel

Voulez-vous une interface intuitive et simple pour votre nouveau projet ? Si la réponse est OUI, un LCD graphique avec dalle tactile est le meilleur choix, parce que ensemble ils forment un écran tactile (LCD graphique + dalle tactile = écran tactile). Ainsi, avec un poignier de composants, vous pouvez réaliser un appareil attractif et facile à utiliser.

Qu'est qu'un écran tactile ? Un écran tactile consiste en une dalle tactile autocollante fixée sur un LCD graphique. Elle est si sensible que même une légère pression suffit pour faire varier le signal de sortie. Il existe plusieurs types de dalles tactiles. Le plus simple est la dalle tactile résistive dont nous allons vous parler ici.

## Principe de fonctionnement

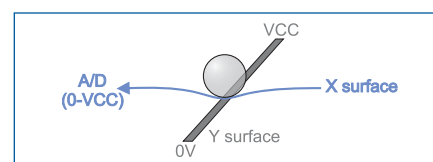
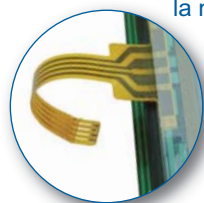
Une dalle tactile résistive consiste en deux films transparents en « sandwich » avec une couche résistive sur les cotés intérieurs des films. La résistance de ces couches ne dépasse en général pas le 1 k $\Omega$ . Les autres cotés des films ont des contacts pour connecter une nappe. L'algorithme pour déterminer la position de la pression se décompose en deux étapes. La première étape est la détermination de la coordonnée X et la deuxième est la détermination de la coordonnée Y de la position. Pour déterminer la coordonnée X, il faut connecter le contact de gauche de la surface X à la masse et le contact de droit à l'alimentation. Un diviseur de tension se crée quand on appuie sur l'écran tactile. La valeur du diviseur est disponible sur le contact inférieur de la surface Y. La tension va de

0 V jusqu'à la valeur de l'alimentation et dépend de la coordonnée X. Si le point de pression est plus près du contact de gauche de la surface X, la tension sera plus près de 0 V. Pour déterminer la coordonnée Y, il faut connecter le contact inférieur de la surface Y à la masse et le contact supérieur à l'alimentation et prélever la tension sur le contact de gauche de la surface X.

## Connecter un microcontrôleur

Pour connecter la dalle tactile à un microcontrôleur, il est nécessaire de réaliser un pilote de dalle tactile. Ce montage permet au microcontrôleur de relier les bons contacts de la dalle tactile à la masse et à l'alimentation (comme décrit plus haut) pour ensuite déterminer les coordonnées X et Y (voir schéma 1). Le contact inférieur de la surface Y et le contact de gauche de la surface X

Détail de la nappe



Détermination de la coordonnée X

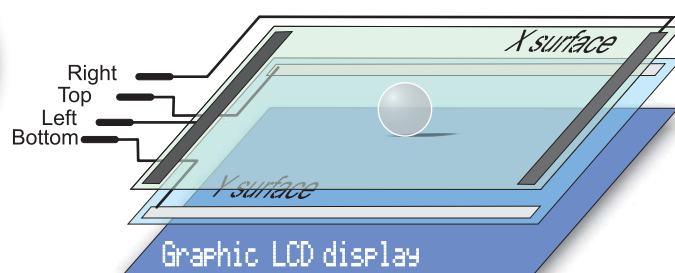
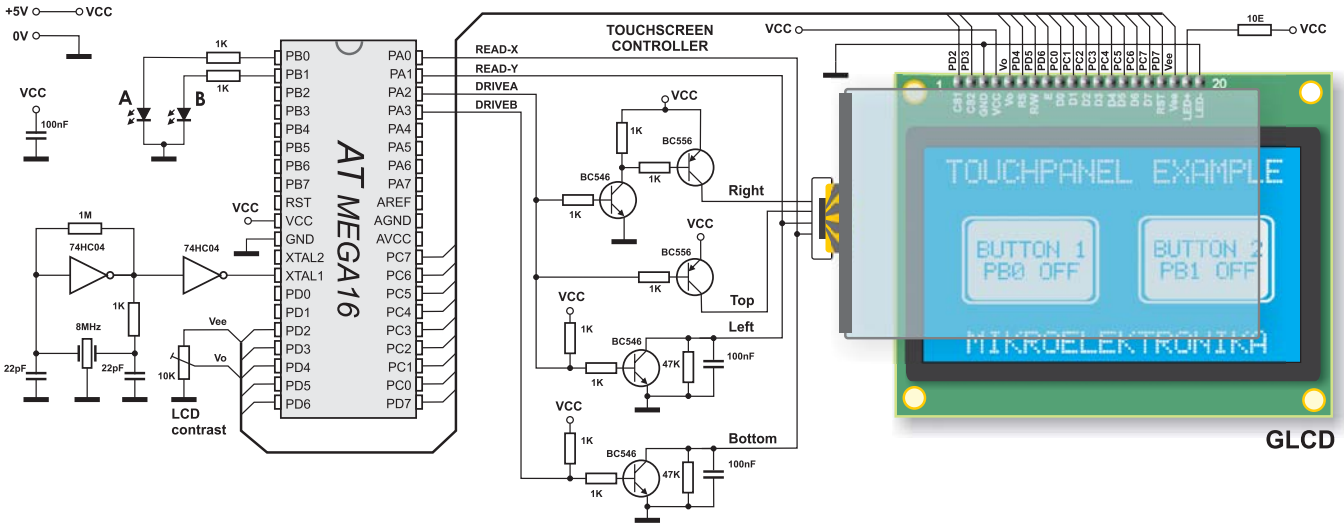


Figure 1. Structure interne de la dalle tactile



sont reliés au convertisseur A/N du microcontrôleur. Les coordonnées X et Y sont déterminés en mesurant les tensions présentes sur ces contacts. Le logiciel doit être capable d'afficher un menu sur le LCD graphique, de contrôler le pilote de la dalle tactile et de lire les valeurs du convertisseur A/N, qui représentent les coordonnées X et Y de la position. Une fois les coordonnées connues, il est possible de décider ce que le microcontrôleur doit faire. Regardez l'exemple 1. Il montre comment activer et désactiver avec un afficheur et une dalle tactile deux broches du microcontrôleur connectées au deux LED A et B.



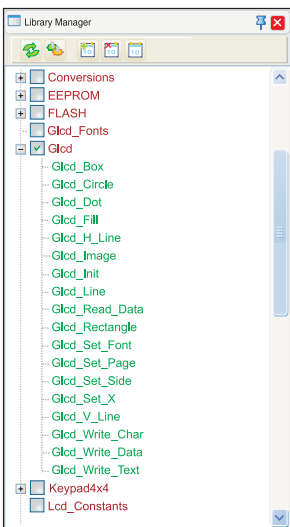
Le connecteur de nappe embarqué avant...



... et après la connexion de l'écran tactile

Vu que la surface de la dalle tactile est légèrement plus grande que la surface du LCD, au cas vous auriez besoin d'une précision plus importante pour déterminer les coordonnées, il est nécessaire de calibrer la dalle tactile avec le logiciel.

### Fonctions utilisées dans ce programme



- ADC\_Read() Lire valeur analogique
  - Delay\_ms() Retard
  - Glcd\_box() Dessiner un rectangle opaque\*
  - Glcd\_circle() Dessiner un cercle
  - Glcd\_Dot() Afficher un point
  - Glcd\_Fill() Effacer/Remplir l'écran\*
  - Glcd\_H\_Line() dessiner une ligne horizontale
  - Glcd\_Image() Importer une image
  - Glcd\_Init() Initialiser l'afficheur LCD\*
  - Glcd\_Line() Dessiner une ligne
  - Glcd\_Read\_Data() Lire des données du LCD
  - Glcd\_Rectangle() Dessiner un rectangle\*
  - Glcd\_Set\_Font() Choisir une police\*
  - Glcd\_Set\_Page() Choisir une page
  - Glcd\_Set\_Side() Choisir la coté de l'afficheur
  - Glcd\_Set\_X() Déterminer la coordonnée X
  - Glcd\_V\_Line() Dessiner une ligne verticale
  - Glcd\_Write\_Char() Écrire un caractère
  - Glcd\_Write\_Data() Écrire des données
  - Glcd\_Write\_Text() Écrire du texte\*
- \* les fonctions de la librairie Glcd

### Exemple 1: Programme montrant le fonctionnement de l'écran tactile

```

program TouchPanel
dim GLCD_DataPort as byte at PORTC
GLCD_DataPort_Direction as byte at DDRC

dim GLCD_CS1 as sbit at PORTD.B2
GLCD_CS2 as sbit at PORTD.B3
GLCD_RS as sbit at PORTD.B4
GLCD_RW as sbit at PORTD.B5
GLCD_EN as sbit at PORTD.B6
GLCD_RST as sbit at PORTD.B7
GLCD_CS1_Direction as sbit at DDRD.B2
GLCD_CS2_Direction as sbit at DDRD.B3
GLCD_RS_Direction as sbit at DDRD.B4
GLCD_RW_Direction as sbit at DDRD.B5
GLCD_EN_Direction as sbit at DDRD.B6
GLCD_RST_Direction as sbit at DDRD.B7

dim DRIVE_A as sbit at PORTA.B2
DRIVE_B as sbit at PORTA.B3
DRIVE_A_Direction as sbit at DDRA.B2
DRIVE_B_Direction as sbit at DDRA.B3

dim x_coord, y_coord, x_coord128, y_coord64 as longint

sub function GetX() as word
DRIVE_A = 1
DRIVE_B = 0
Delay_ms(5)
result = ADC_Read(0)
end sub

sub function GetY() as word
DRIVE_A = 0
DRIVE_B = 1
Delay_ms(5)
result = ADC_Read(1)
end sub

main:
DRIVE_A_Direction = 1
DRIVE_B_Direction = 1
PORTB.B0 = 0
PORTB.B1 = 0
PORTB.B1 = 1

Glcd_Init()
Glcd_Set_Font(font5x7, 5, 7, 32)
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)
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 TRUE
x_coord = GetX()
y_coord = GetY()
x_coord128 = (x_coord * 128) / 1024
y_coord64 = 64 - (y_coord * 64) / 1024
if (x_coord128 >= 10) and (x_coord128 <= 58) and (y_coord64 >= 18) and (y_coord64 <= 46) then
if (PORTB.B0 = 0) then
PORTB.B0 = 1
Glcd_Write_Text("PB0 ON";14,4,0)
else
PORTB.B0 = 0
Glcd_Write_Text("PB0 OFF";14,4,0)
end if
end if
if ((x_coord128 >= 70) and (x_coord128 <= 118) and (y_coord64 >= 18) and (y_coord64 <= 46) then
if (PORTB.B1 = 0) then
PORTB.B1 = 1
Glcd_Write_Text("PB1 ON";74,4,0)
else
PORTB.B1 = 0
Glcd_Write_Text("PB1 OFF";74,4,0)
end if
end if
Delay_ms(100)
wend
while true
end
    
```

**mikroBASIC PRO pour AVR® Éditeur de librairie avec des librairies prêtes à l'emploi, comme Ethernet, CAN, SD/MMC, etc.**

**NOTE:** Les codes source pour microcontrôleurs AVR de cet exemple écrit en C, BASIC et PASCAL ainsi que tous les programmes écrits pour les microcontrôleurs PIC® et dsPIC® sont disponibles sur notre site Internet [www.mikroe.com/en/article/](http://www.mikroe.com/en/article/)

