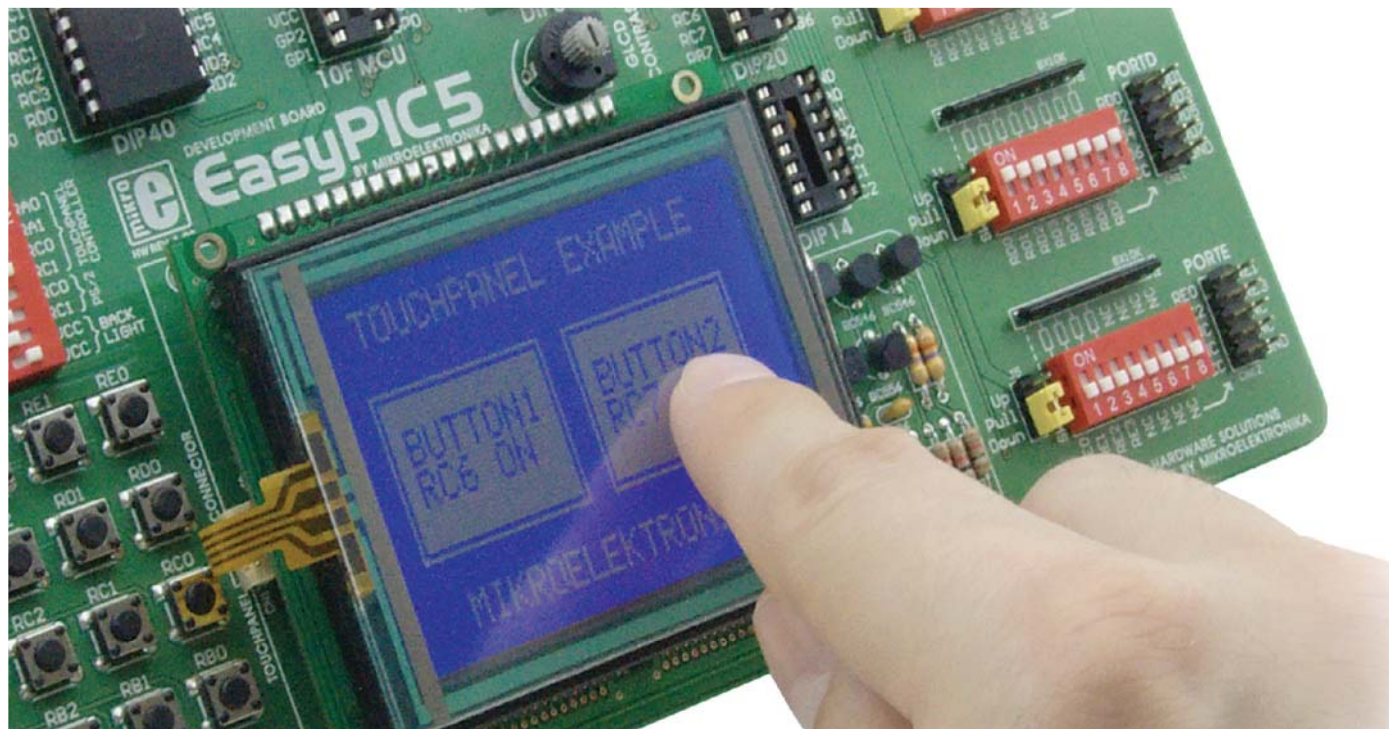


Bon. Maintenant il vous faut un... écran tactile



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

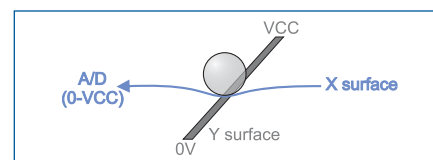
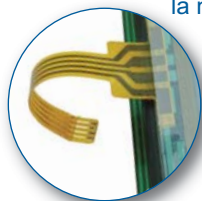
Une dalle tactile résistive consiste en deux films transparents en « sandwich » avec une couche résistive sur les cotés intérieures des films. La résistance de ces couches ne dépasse en général pas le 1 k Ω . Les autres cotés des films ont des contacts pour connecter une nappe. L'algorithme pour déterminer 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

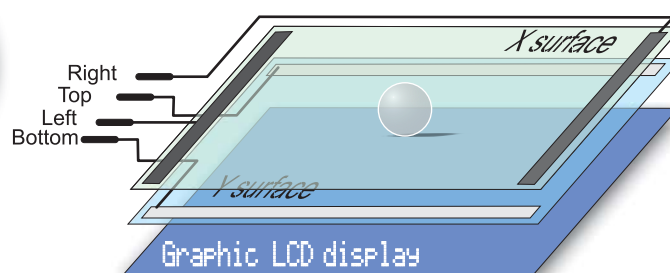
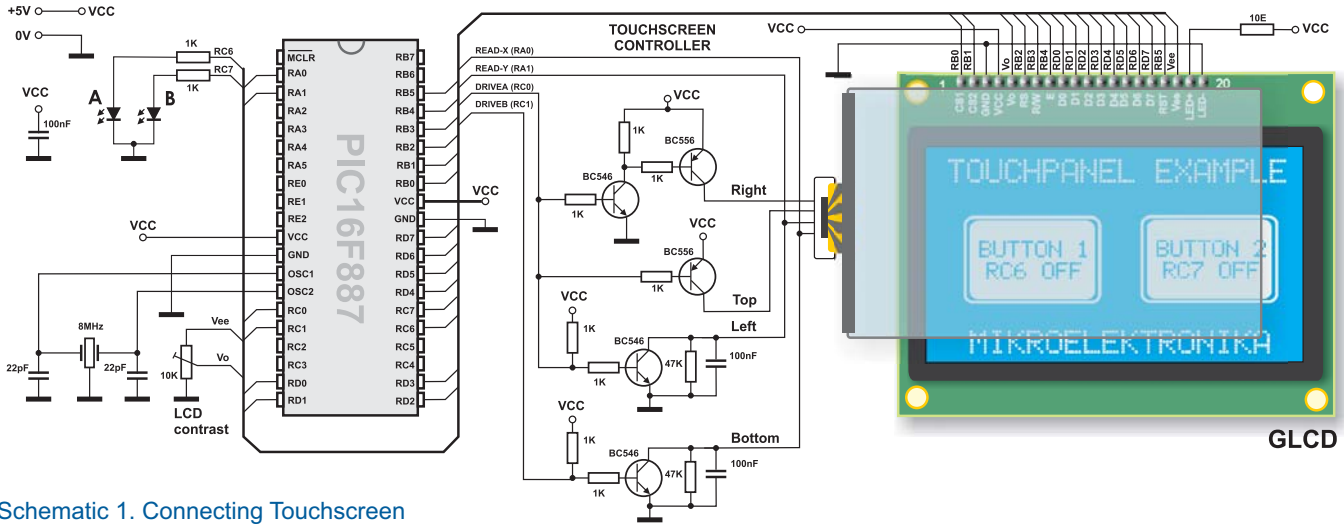


Figure 1. Structure interne de la dalle tactile



Schematic 1. Connecting Touchscreen

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.

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

```
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)
//DRIVEB = 0 (BOTTOM drive off)
Delay_ms(5);
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)
//DRIVEB = 1 (BOTTOM drive on)
Delay_ms(5);
return ADC_read(1); // reading Y value from RA1 (from LEFT)
}

void main() {
PORTA = 0x00; // RA0 i RA1 are analog inputs
TRISA = 0x03; // Configure other AN pins as digital I/O
ANSEL = 0x03;
ANSELH = 0;

PORTC = 0; // PORTC is output
TRISC = 0;

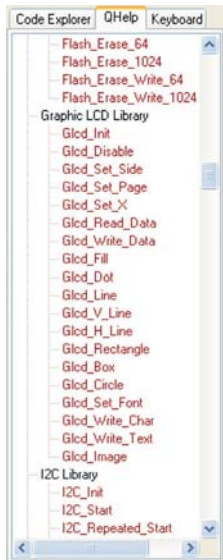
Glcd_Init(&PORTB, 0, 1, 2, 3, 5, 4, &PORTD); // Glcd_Init_EP5
Glcd_Set_Font(FontSystems5x8, 5, 8, 32); // Choose font
Glcd_Fill(0); // Clear GLCD

CopyConst2Ram(msg,msg1); // Copy "TOUCHPANEL EXAMPLE" string to RAM
Glcd_Write_Text(msg,10,0,1); // Glcd_Write_Text(msg,10,0,1);
CopyConst2Ram(msg,msg2); // Copy "MIKROELEKTRONIKA" string to RAM
Glcd_Write_Text(msg,17,7,1); // 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); // Glcd_Write_Text(msg,14,3,0);
CopyConst2Ram(msg,msg5); // Copy "RC6 OFF" string to RAM
Glcd_Write_Text(msg,14,4,0); // Glcd_Write_Text(msg,14,4,0);
CopyConst2Ram(msg,msg4); // Copy "BUTTON2" string to RAM
Glcd_Write_Text(msg,74,3,0); // Glcd_Write_Text(msg,74,3,0);
CopyConst2Ram(msg,msg6); // Copy "RC7 OFF" string to RAM
Glcd_Write_Text(msg,74,4,0); // 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); // 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); // 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); // 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); // Glcd_Write_Text(msg,74,4,0);
}
}
Delay_ms(100);
}
}
```



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_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

mikroC pour PIC® Éditeur de librairie avec des librairies prêtes à l'emploi, comme Ethernet, CAN, SD/MMC, etc.

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

