

# Ćwiczenie 4

## Panel dotykowy

### Bibliografia

- [1] Karta katalogowa mikrokontrolera LM4F232H5QD - <http://www.ti.com/lit/ds/symlink/tm4c123gh6pge.pdf>
- [2] Instrukcja obsługi płyty EasyMxPROv7 [http://www.mikroe.com/downloads/get/1812/easymx\\_pro\\_v7\\_stellaris\\_manual\\_v102.pdf](http://www.mikroe.com/downloads/get/1812/easymx_pro_v7_stellaris_manual_v102.pdf)
- [3] Strona wiki Texas Instruments dotycząca płyty EasyMxPROv7 <http://processors.wiki.ti.com/index.php/EasyMxPRO>
- [4] Instrukcja obsługi ROM LM4F232H5QD <http://www.ti.com/lit/ug/spmu270a/spmu270a.pdf>
- [5] Instrukcja obsługi biblioteki TivaWare Graphics Library <http://www.ti.com/lit/ug/spmu300a/spmu300a.pdf>

### Wymagania:

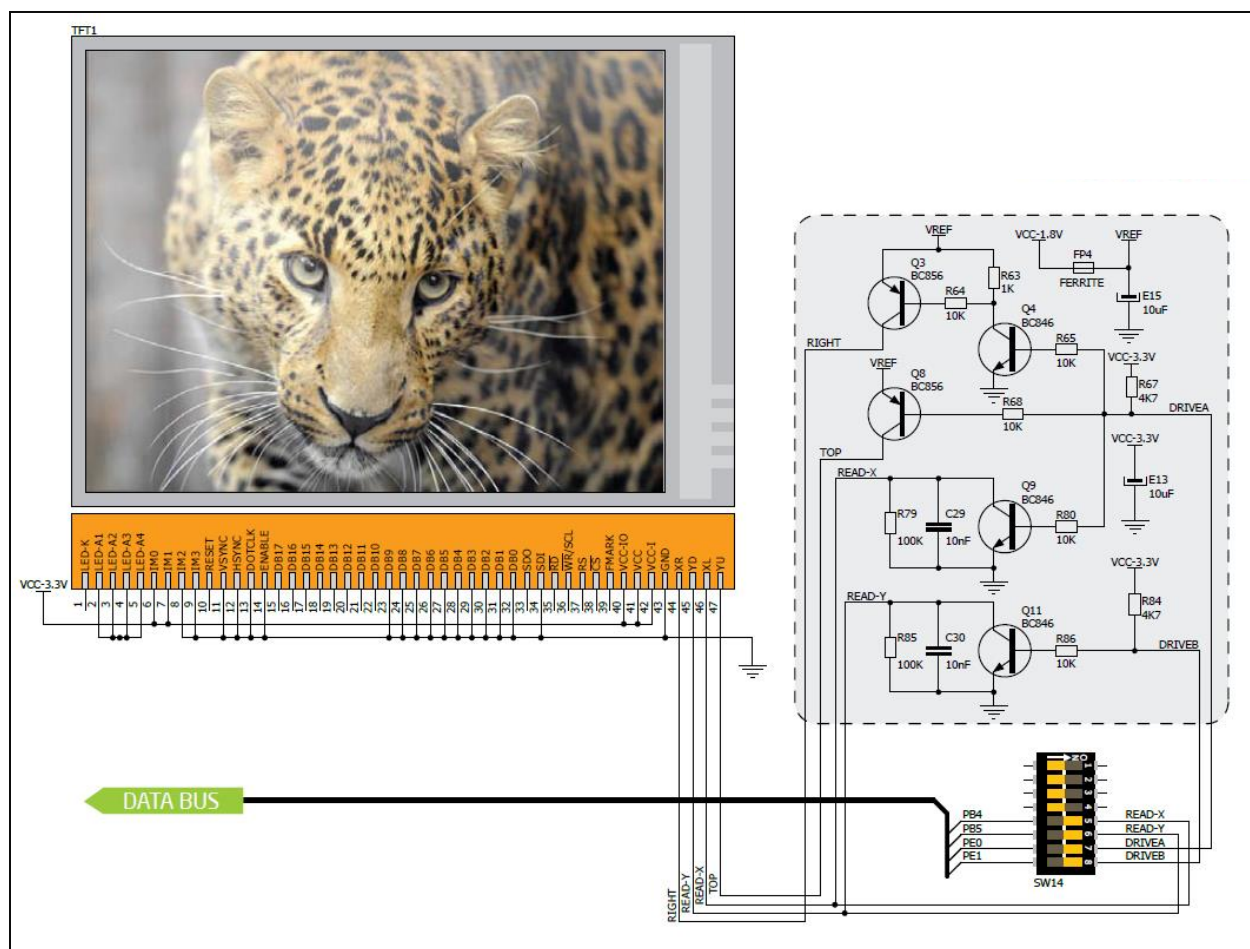
- Znajomość środowiska CodeComposerStudio, umiejętność utworzenia projektu, zarządzania, uruchomienia, ustawienia opcji linkera oraz assemblera
- Znajomość podstawowych funkcji z klasy GPIO
- Ogólna wiedza dotycząca biblioteki TivaWare Graphics Library - baza teoretyczna

### Wprowadzenie, ustawienie płyty EasyMX Pro v.7

Generalnie, panel dotykowy to panel szklany, którego powierzchnia jest pokryta dwoma warstwami rezystywnego materiału. Dotykając panelu zwiększamy ciśnienie w danym punkcie, które jest mierzone przez dedykowany kontroler, i na tej podstawie może zostać ustalona lokalizacja naciśnięcia panelu. W przypadku płyty EasyMX Pro v7 mamy do czynienia z panelem sterowanym za pomocą czterech linii: READ-X, READ-Y, DRIVEA oraz DRIVEB. Wbudowany kontroler pozwala na pomiar siły nacisku w dowolnym punkcie, reprezentując współrzędne naciśniętego punktu w postaci wartości napięcia analogowego. Wartość ta z kolei może być bezpośrednio przetworzona do czystych współrzędnych X oraz Y na wyświetlaczu.

Aby umożliwić korzystanie z panelu dotykowego, należy zezwolić na podłączenie do mikroprocesora czterech wymienionych wyżej sygnałów za pomocą przestawienia odpowiednich przełączników w grupie SW14. I tak, przełącznik SW14.5 łączy sygnał READ-X z wejściem analogowym PB4, SW14.6 łączy sygnał READ-Y z wejściem analogowym PB5, SW14.7 łączy sygnał DRIVEA z wejściem cyfrowym PE0, oraz SW14.8 łączy sygnał DRIVEB z wejściem cyfrowym PE1.

Schemat połączenia panelu do mikroprocesora przedstawiony został na poniższym rysunku:

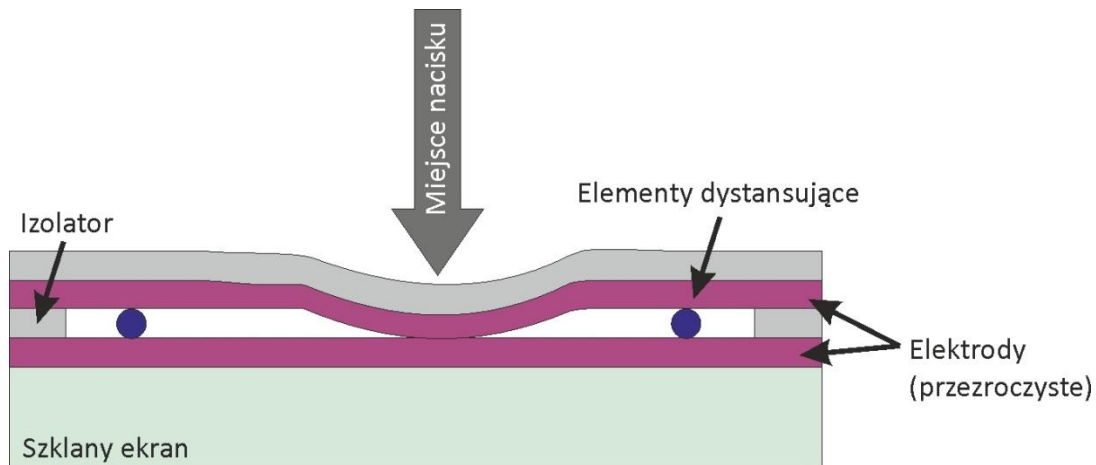


Rysunek 1. Schemat podłączenia panelu dotykowego do mikrokontrolera

## Technologia i rodzaje paneli dotykowych

Panele dotykowe są powszechnie wykorzystywane w wielu dziedzinach życia, poczynając od smart fonów, tabletów, poprzez nawigacje samochodowe, na biletomatach, bankomatach czy kserokopiarkach kończąc. W zależności od zastosowania dany panel dotykowy jest wykonany w odpowiadającej temu zastosowaniu technologii. Wyróżnić tutaj możemy:

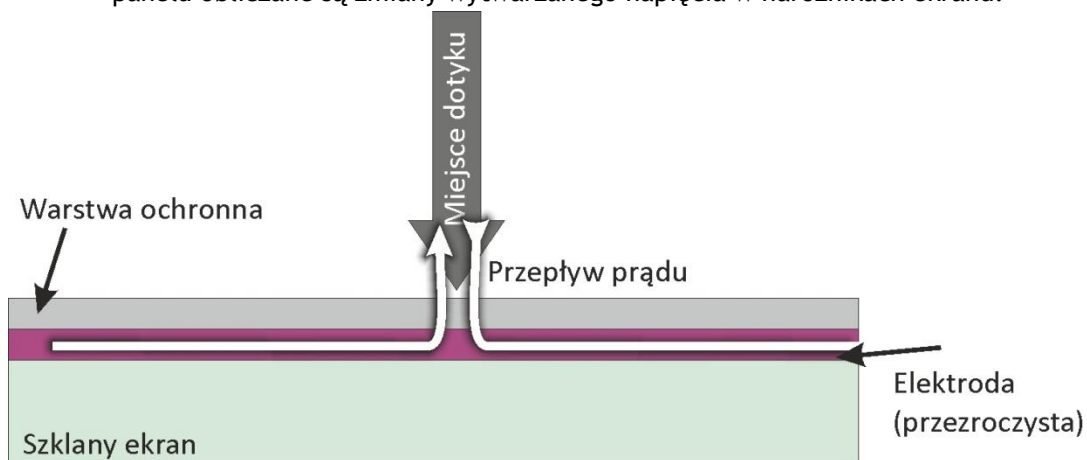
- panele rezystywne (opornościowe, rezystancyjne)  
w tym przypadku ekran LCD jest przykryty dwoma przezroczystymi elektrodami (wykonanymi z tlenku indowo-cynowego (indium-tin-oxide, ITO)), przedzielonymi pustą przestrzenią z odpowiednimi elementami dystansowymi i izolatorami. Na samym wierzchu natomiast umieszczona jest elastyczna warstwa wykonana np. z polietylenu (PET). W wyniku dotknięcia panelu wierzchnia warstwa ugina się zezwalając na zmianę napięcia w punkcie dotknięcia, mierzoną za pomocą elementów na brzegach wyświetlacza. Zmiana ciśnienia (nacisku palcem czy stylusem) na napięcie, a następnie na współrzędne pozwala na wyznaczenie punktu dotyku panelu.



Rysunek 2. Ilustracja działania panelu rezystywnego

- panele pojemnościowe
  - W odróżnieniu od panelu rezystancyjnych, w panelach pojemnościowo-powierzchniowych mierzona jest zmiana pojemności powstała wskutek dotknięcia palcem. Panele te nie rozpoznają przez to dotknięcia palcem, gdy założona jest rękawiczka, czy też na typowe proste stylusy.
    - panele pojemnościowo-powierzchniowe
 

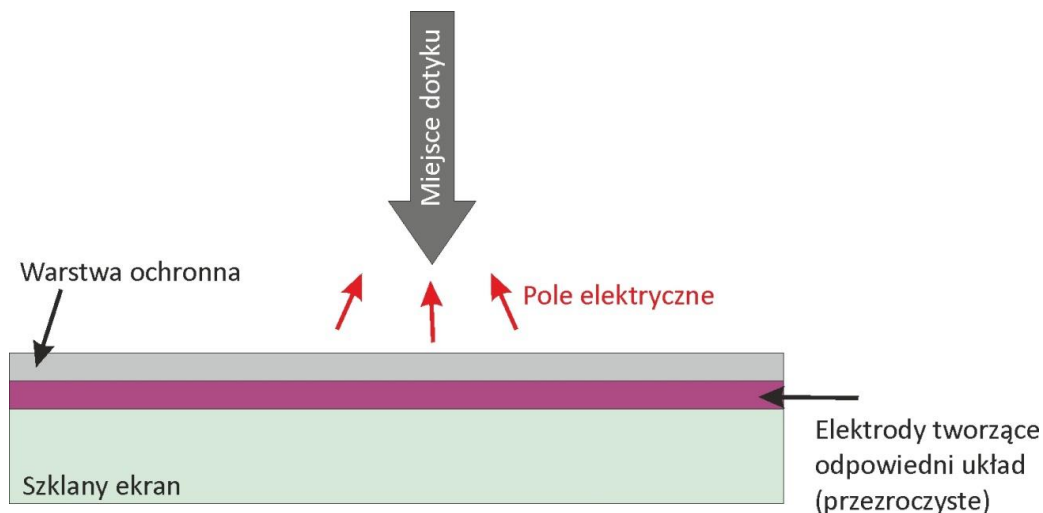
Przykładowym rozwiązaniem jest umieszczenie czterech elektrod w narożnikach wyświetlacza generujących jednolite napięcie powierzchniowe, w wyniku dotknięcia panelu obliczane są zmiany wytwarzanego napięcia w narożnikach ekranu.



Rysunek 3. Ilustracja działania panelu pojemnościowo powierzchniowego

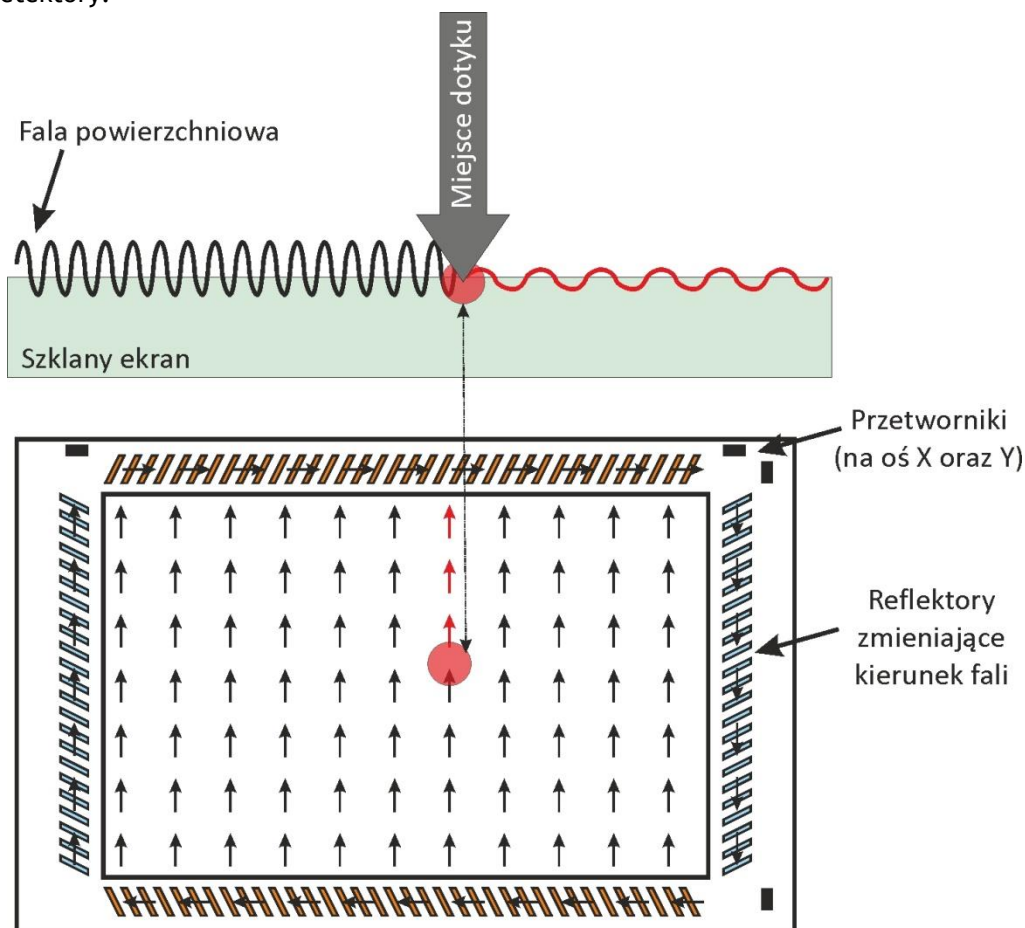
- panele pojemnościowo-projekcyjne
 

W tym przypadku panel jest wyposażony w gęstą siatkę elektrod pozwalających na precyzyjne określenia miejsca dotyku, a także na wykrywanie wielu punktów dotyku. Jednym z rozwiązań jest stosowanie siatek z tlenku indowo-cynowego (indium-tin-oxide, ITO).



Rysunek 4. Ilustracja działania panelu pojemnościowo projekcyjnego

- panele wykorzystujące efekt akustycznej fali powierzchniowej  
W tym rozwiązaniu w narożnikach panelu umieszczone są przetworniki piezoelektryczne generujące akustyczną falę powierzchniową (Surface Accoustic Wave, SAW). W wyniku dotknięcia ekranu następuje zniekształcenie i tłumienie fali, mierzone przez odpowiednie detektory.



Rysunek 5. Ilustracja działania panelu wykorzystującego akustyczną falę powierzchniową

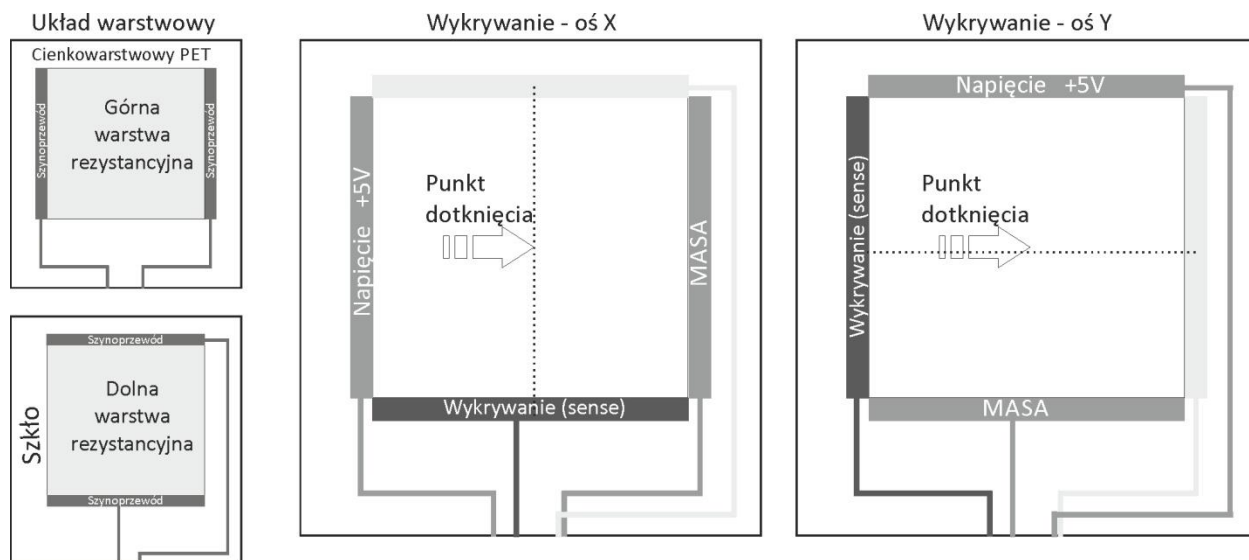
- Panele wykorzystujące podczerwień (Infrared optical imaging)
- Panele wykorzystujące indukcję magnetyczną

## Kontroler panelu dotykowego w płycie EasyMX Pro v7

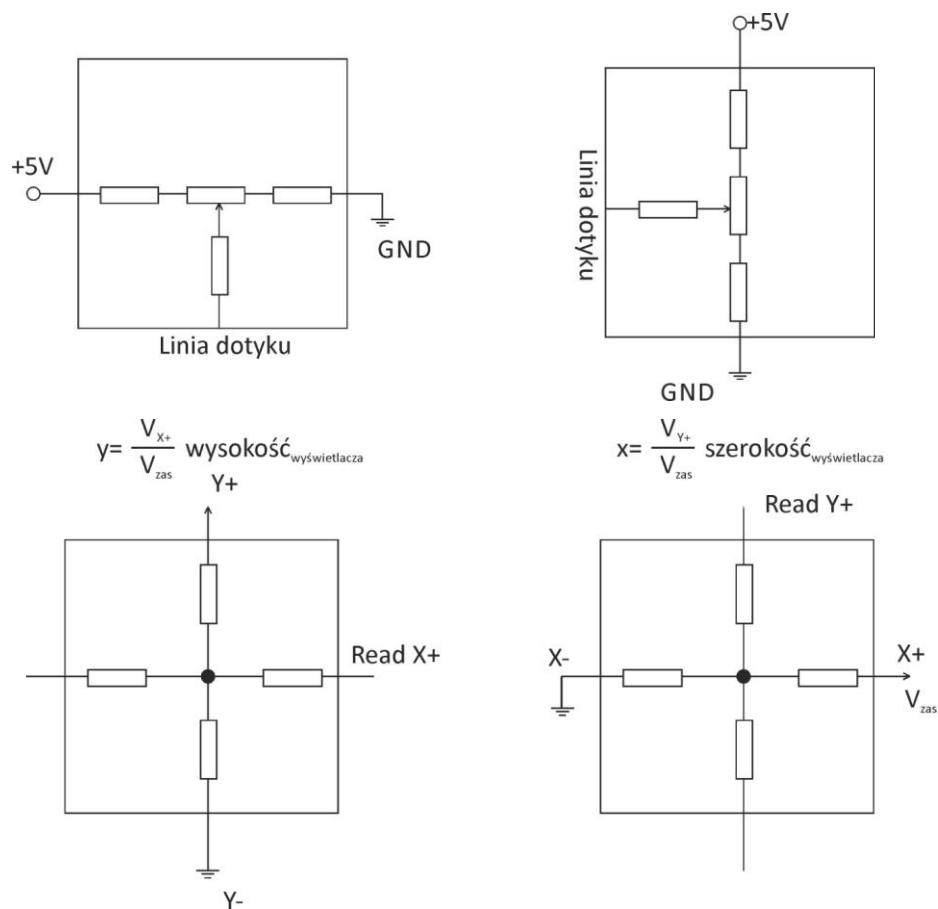
W przypadku płyty EasyMX PRO v7 zastosowany został panel rezystywny sterowany czterema liniami. Zasada działania takiego panelu została przedstawiona na rysunkach 14 i 15.

Każda warstwa rezystancyjna jest podłączona do sterownika z dwóch przeciwległych stron, przykładowo górna warstwa jest podłączona stronami lewą i prawą, zaś dolna - górną i dolną. Sterownik cyklicznie, z dużą częstotliwością, dostarcza wiele razy na sekundę napięcie do odpowiednich elementów obu warstw umożliwiając przez to odczyt położenia. Bazując na Rysunku 6 można zauważyć, że w pierwszym etapie napięcie +5V jest dostarczone górnej warstwy panelu, a w szczególności do jego lewej części, pozostawiając prawą część połączoną z masą (np. 0V). W wyniku dotknięcia panelu powstaje dzielnik rezystancyjny, pozwalający na odczyt położenia X na podstawie zmierzonego napięcia (rysunek 6) na warstwie dolnej. W drugiej fazie role warstw zostają zamienione, napięcie zostaje przyłożone do górnej części panelu w warstwie dolnej, natomiast odczyt położenia Y następuje na podstawie pomiaru napięcia w warstwie górnej.

Analizując schemat z rysunku 1 i 6 można zauważyć, że panel dotykowy wyświetlacza komunikuje się poprzez cztery linie: XL oraz XR, a także YU oraz YD, podłączone odpowiednio do linii READ-X, RIGHT, TOP, READ-Y. Oczywiście, XL i XR oznaczają odpowiednio stronę lewą i prawą (left, right), zaś YU i YD - górę i dół (up, down) wyświetlacza.



Rysunek 6. Ilustracja pomiaru położenia na panelu dotykowym



Rysunek 7. Obrazowy schemat obwodów w momencie pomiaru położenia

## Program 1

W celu zapoznania się z podstawowymi zasadami działania panelu dotykowego należy zaimportować do przestrzeni roboczej program *TouchScreen\_pure.zip*. Program ten jest przygotowany do uruchomienia na płycie ewaluacyjnej, w zależności jednak od ustawień systemowych może okazać się konieczne wskazanie (poprawienie) ścieżek, wskazujących na położenie poszczególnych bibliotek. Program wyświetla współrzędne punktu dotkniętego przez użytkownika.

**UWAGA:** Aby skorzystać z funkcjonalności panelu dotykowego należy jednocześnie wykorzystać moduł ADC a także przerwania. Zarówno moduł ADC, jak i system przerwań, będą omawiane szczegółowo w następnych ćwiczeniach. Celem tego ćwiczenia jest skupienie się na działaniu panelu dotykowego.

Spośród wielu plików wchodzących w skład projektu *TouchScreen\_pure* należy zwrócić uwagę na następujące:

- *main.c* - plik, w którym znajduje się główny program
- *startup\_ccs.c* - plik służący do obsługi przerwań
- *drivers/ili9341\_240x320x262K.c* - znany już plik z funkcjami pozwalającymi na korzystanie z wyświetlacza
- *drivers/touch.c* - plik zawierający podstawowe funkcje służące do obsługi panelu dotykowego

Analizując kod źródłowy w pliku `main.c` można zauważyć trzy linie istotne z punktu widzenia działania wyświetlacza:

```
TouchScreenInit();  
TouchScreenCalibrate(&sContext);  
TouchScreenCallbackSet(TouchCallback);
```

Pierwsza komenda inicjalizuje panel dotykowy (definicja w pliku `touch.c`). Druga komenda uruchamia procedurę kalibracji wyświetlacza. Trzecia zaś uaktywnia przerwanie, tzn. w momencie wykrycia przerwania pochodzącego od panelu dotykowego uruchamiana jest funkcja `TouchCallback`, zdefiniowana w pliku `main.c`. Warto zwrócić uwagę, że w rzeczywistości przerwanie wywoływane jest przez układ ADC, który zmierzył wartość napięcia i fakt ten sygnalizuje wystawieniem przerwania.

## Zadanie 1

---

Napisać program, będzie wskazywał współrzędne ostatnio naciśniętego na wyświetlaczu punktu dokładnie w tym punkcie (tzn. jak naciśniemy wyświetlacz w lewym dolnym rogu to informacja o miejscu dotknięcia powinna pojawić się też w lewym dolnym rogu).

## Zadanie 2

---

Należy przeanalizować, co dokładnie jest wykonywane w funkcjach `TouchScreenInit` oraz `TouchScreenCalibrate`. W szczególności należy sprawdzić, jaki jest cel stosowania i znaczenie zmiennych `g_floTouchCalibrationA`, `g_floTouchCalibrationB`, `g_floTouchCalibrationC` oraz `g_floTouchCalibrationD`. Należy zmodyfikować program w taki sposób, aby można było pominąć wywołanie funkcji służącej do kalibracji (funkcja `TouchScreenCalibrate` nie ma być wywoływana; w ogóle w funkcji `main` nie powinno być nic związane z kalibracją wyświetlacza, to zadanie dla sterowników). Proszę pominąć kalibrację w kolejnych zadaniach.

### UWAGA:

Cele kalibracji mogą być wielorakie. W naszym przypadku kalibracja ma na celu ustalenie sposobu konwersji napięcia pobranego z panelu dotykowego na pozycję na wyświetlaczu (np. nr piksela). Wyświetlacz jest bowiem niezależnym modulem względem panelu dotykowego - zdarzenie przechwycone przez panel dotykowe musi być następnie przetworzone i odpowiednio użyte w wyświetlaczu LCD.

W celu kalibracji wywołana jest funkcja:

```
void TouchScreenCalibrate(tContext *psContext)
```

Wartość napięcia pobrana z dzielnika napięcia (Rys. 6 oraz 7) zostaje przetworzona w układzie ADC i zapisywana w tablicy `M`

```
M[i][X] = g_i16TouchX;  
M[i][Y] = g_i16TouchY;
```

Jednocześnie w tablicy `E` są przechowywane indeksy pikseli, “na których” znajduje się środek wyświetlanego krzyżyka. Wartości zapisane w tablicy `M` mogą uzyskiwać wartości nawet do 2000. W celu przetworzenia uzyskanych z ADC wartości na odpowiadające im położenie na wyświetlaczu utworzona jest zmienna:

```
g_floTouchCalibrationA = ((E[1][X]-E[0][X])) / (M[1][X]-M[0][X]);
```



której jednostka ilustruje relację szerokości ekranu do uzyskanego napięcia z panelu dotykowego np. [piksel]/[Volt]. Jest to więc jakby współczynnik skalujący. Po obliczeniu wartości tej zmiennej ustalana jest jeszcze wartość stałego przesunięcia.

$$g\_floTouchCalibrationB = E[0][X] - (M[0][X] * g\_floTouchCalibrationA);$$

Operacja jest powtarzana dla drugiego wymiaru.

$$g\_floTouchCalibrationC = ((E[1][Y]-E[0][Y])) / (M[1][Y]-M[0][Y]);$$
$$g\_floTouchCalibrationD = E[0][Y] - (M[0][Y] * g\_floTouchCalibrationC);$$

Obliczone wartości zmiennych `g_floTouchCalibration` są później wykorzystywane podczas obliczania wartości piksela odpowiadającego dotkniętemu miejscu na panelu dotykowym:

```
//  
// Convert the ADC readings into pixel values on the screen.  
//  
g_i16X = (int16_t) ((g_i16TouchX * g_floTouchCalibrationA) +  
                   g_floTouchCalibrationB);  
g_i16Y = (int16_t) ((g_i16TouchY * g_floTouchCalibrationC) +  
                   g_floTouchCalibrationD);
```

### Zadanie 3

---

Należy zmodyfikować program w taki sposób, aby wyświetlał współrzędne położenia:

- względem środka ekranu
- podawał je w centymetrach (calach) względem lewego górnego narożnika.

### Zadanie 4

---

Napisać program będący rozszerzeniem programu „MENU” z zadania 4. Do poprzedniego programu należy dodać możliwość wyboru dowolnego pola za pomocą panelu dotykowego.

UWAGA:

Należy zwrócić uwagę, że każdorazowo w przypadku dotknięcia panelu dotykowego wywoływane są dwa, kolejno po sobie następujące przerwania odpowiadające dwóm sytuacjom: a) dotknięcia i b) puszczenia panelu. Informacja na temat bieżącej sytuacji wywołującej przerwanie jest określona za pomocą wiadomości zdefiniowanych w pliku nagłówkowym `widget.h`. W pliku tym można odnaleźć następujące definicje:

- `#define WIDGET_MSG_PTR_DOWN x00000002` - wiadomość generowana w przypadku dotknięcia panelu
- `#define WIDGET_MSG_PTR_UP x00000004` - wiadomość generowana w przypadku puszczenia panelu

W dostarczonym przykładzie do przerwania przekazywana jest każdorazowo informacja na temat sytuacji wywołującej przerwanie. Przykładowe wykorzystanie tej informacji może być następujące:

- `if (ulMessage == WIDGET_MSG_PTR_DOWN) ....`

### Zadanie 5

---

Napisać program, w którym użytkownik będzie mógł rysować palcem po wyświetlaczu. Proponowana funkcjonalność - kilka opcji do wyboru, choć można wszystko: wybór koloru rysowania, wybór grubości linii, opcja gumki, wybór kształtu rysika/pędzla. Dodatkowo, zmiana kolorów i grubości może odbywać



się liniowo (np. z krokiem co 1), albo użytkownik ma do wyboru kilka ustalonych wartości. Wybór może odbywać się z użyciem albo przycisków (np. Port A), albo za pomocą ekranu dotykowego.

## Zadanie 6

---

Napisać program rozpoznający dwa (lub w wersji rozszerzonej cztery) dowolne gesty, np. przesunięcie palcem z dołu do góry w kierunku ukośnym w prawo, przesunięcie palcem z góry do dołu w kierunku ukośnym w prawo. W zależności od rozpoznanego gestu powinien się na wyświetlaczy pojawić odpowiedni napis.

## Zadanie 7

---

Napisać program, w którym użytkownik będzie mógł palcem przesuwając po ekranie uprzednio narysowany obiekt (np. prostokąt).