

Lab 5. Wyświetlacz LCD i ekran dotykowy

Cele :

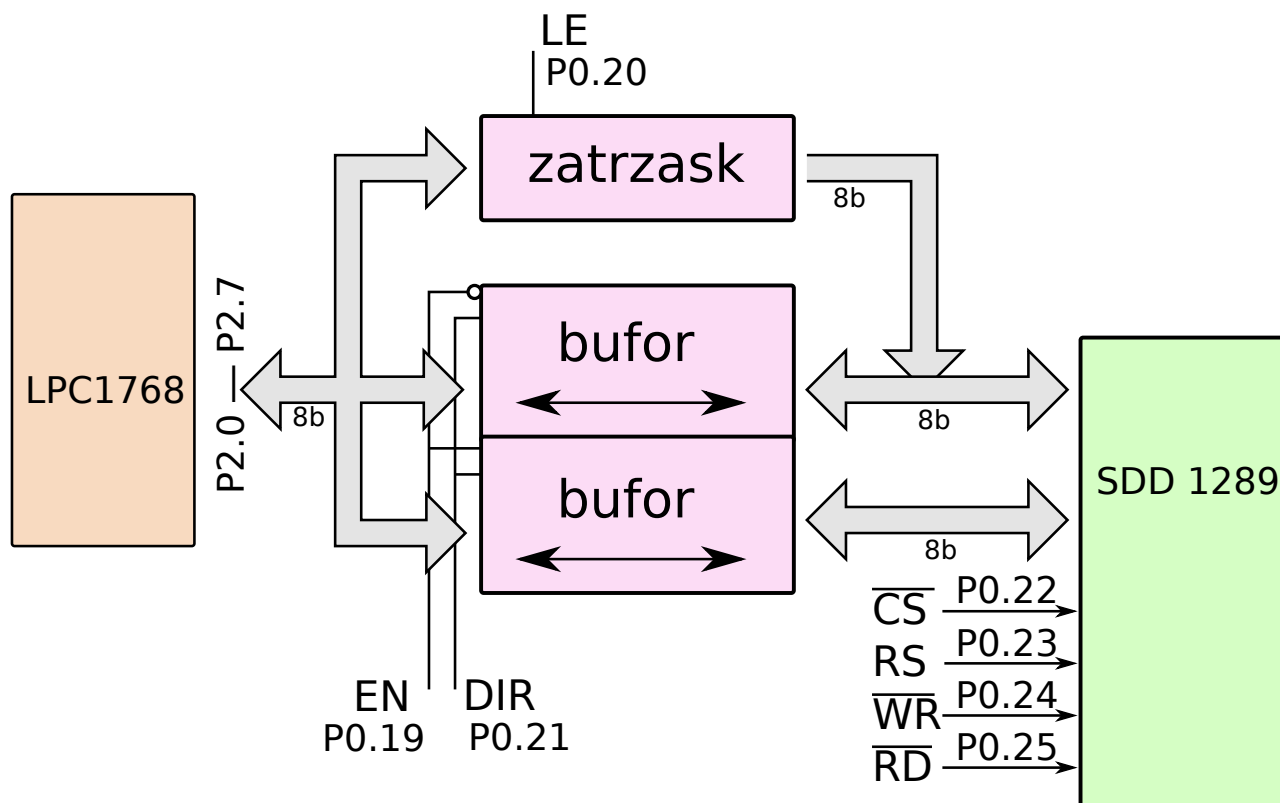
- dokładanie bibliotek spoza Keila,
- współpraca ze sterownikiem ekranu LCD,
- specyfika rezystancyjnego ekranu dotykowego,
- interakcja pomiędzy LCD i ekranem dotykowym.

Kroki :

1. Komentarz prowadzącego nt. podłączenia sterownika wyświetlacza do LPC1768 (patrz dodatek B).
2. Ściągnij archiwum i dodaj bazę do projektu
 - Ściągnij archiwum `lcd_lib.tar.gz` z strony przedmiotu (sekcja Laboratorium, podstrona "Ekran LCD+TP dla Open 1768")
 - Otwórz i przeczytaj `Readme`
 - Dodaj do projektu plik bazowy naszej płyty
3. Konfiguracja portów i ustalenie podłączonego sterownika (potencjalnie mamy ILI9325 lub SDD1289)
 - Przeglądaj procedury z plików `Open1768_LCD` i spróbuj zrozumieć zależności między nimi
 - Skonfiguruj porty w mikrokontrolerze wywołując odpowiednią funkcję z biblioteki
 - Przeczytaj `DeviceCode` sterownika wyświetlacza z jego rejestru `OSCIL_ON` (numer `0x00`)
 - Na podstawie odczytanej wartości zdecyduj który sterownik jest właściwy ILI9325 czy SDD1289.
4. Inicjacja wyświetlacza i zapalenia piksela
 - Na podstawie poprzedniego punktu dołącz plik układu sterującego wyświetlaczem (sterownika) i zdefiniuj odpowiedni symbol kompilatora (zerknij do `Readme`)
 - Wywołaj procedurę konfigurującą wyświetlacz (jest tylko jedna w nowym nagłówku)
 - Korzystając z wiadomości z wykładu zapal jeden lub kilka pikseli (numery rejestrów i stałe definiujące podstawowe kolory są w pliku nagłówkowym)
5. Szybkie rysowanie tła
 - Napisz procedurę zamazującą cały ekran jednolitym kolorem (może być też jakiś wzór)
 - Zoptymalizuj rysowanie tła korzystając z autoinkrementacji i wyjmując ustawianie współrzędnych X i Y z pętli; ustawiamy współrzędne początkowe tylko raz na początku
 - Wykonaj drugi krok optymalizacji rozbijając procedurę `lcdWriteReg` na składowe `lcdWriteIndex` i `lcdWriteData`; tylko jedna z nich powinna zostać w pętli
6. Rysowanie prostych figur
 - Napisz procedurę rysującą linię wykorzystując np. algorytm Bresenhama – patrz wikipedia
 - Napisz procedury rysujące kilka prostych figur np. prostokąt, trójkąt, itp.
 - Opcjonalnie można skorzystać z okna autoinkrementacji
 - Opcjonalnie można też narysować okrąg algorytmem Bresenhama
7. Pisanie tekstu
 - Dodaj następny element z biblioteki – plik `asciiLib` z czcionkami i prostym interfejsem dostępowym
 - Spróbuj zgadnąć jak są zakodowane znaki (czcionka rastrowa, monochromatyczna) lub zapytaj prowadzącego
 - Napisz procedurę rysującą jeden znak
 - Napisz procedurę piszącą tekst
 - Opcjonalnie można spróbować zawijać wiersze i/lub przeskalować fonty
8. Komentarz prowadzącego nt. działania i podłączenia ekranu dotykowego (ekran rezystancyjny skomunikowany przez SPI)

9. Odpal ekran dotykowy
 - Ściągnij archiwum `tp_intro.tar.gz` z tego samego miejsca co wcześniej pliki do wyświetlacza
 - Zainicjuj transmisję z ekranem dotykowym (jest tylko jedna funkcja do tego w plikach)
 - Odczytuj pozycję rysika w pętli i wyświetlaj ją gdzieś, aby przekonać się jak funkcjonuje układ współrzędnych ekranu dotykowego
 - Opcjonalnie zaimplementuj procedurę uśredniającą pozycję
10. Opcjonalnie połącz ekran dotykowy z LCD w prostego *Paint*'a
 - Narysuj krzyżyki (punkty) w rogach ekranu LCD (to będą znane miejsca)
 - Ustal jakim współrzędnym ekranu dotykowego odpowiadają twoje punktu na LCD
 - Wyznacz transformację liniową pomiędzy ekranem dotykowym a LCD
 - Zrób prostą aplikację w której można rysować po ekranie LCD

B. Wyświetlacz SDD1289 na płycie LandTiger



Wyświetlacz SDD1289 ma rozdzielczość 320x240. Programujemy go przez jego rejestry wewnętrzne. Sygnał RS rozróżnia czy dane na szynie wejściowej to adres rejestru (1) czy wartość (0). Typowo rejestr zapisujemy używając dwu procedur `LCD_WriteIndex` wysyłając numer rejestru i zaraz potem `LCD_WriteData`. Analogicznie przy odczycie mamy parę `LCD_WriteIndex` następnie `LCD_ReadData`.

Kolor jest 16. bitowy wg wzorca {R (5b), G (6b), B(5b)}. Współrzędne aktualnego punktu w którym zostanie wyświetlony piksel znajdują się w rejestrach `ADRY_RAM` (0x004f) i `ADRX_RAM` (0x004e). Kolor piksela należy wysłać do rejestru `DATA_RAM` (0x0022).

Wyświetlacz ma standardowo włączoną auto-inkrementację pozycji piksela, więc zamazując cały wyświetlacz wystarczy wysłać tylko jedną wartość koloru do rejestru `DATA_RAM` bez pozycji pikseli. Co więcej numer rejestru (indeks) ustawiamy tylko raz a potem używamy tylko funkcji `LCD_WriteData`.

Auto-inkrementacja jest możliwa także w ograniczonym obszarze prostokątnym, którego pozycje przechowują rejestry `HADRPOS_RAM`, `VADRPOS_RAM_START` i `VADRPOS_RAM_END`. Pierwszy zawiera ograniczenia w poziomie a pozostałe dwa w pionie. Domyślnie ograniczenia obejmują cały wyświetlacz.

Auto-inkrementację można także wykorzystać do rysowania linii bowiem można sterować kierunkiem i kolejnością dodawania tzn. najpierw współrzędna X lub najpierw współrzędna Y, zwiększamy współrzędną o jeden lub zmniejszamy (niezależnie w każdej osi). Zachowaniem auto-inkrementacji zarządza rejestr `ENTRYM` (0x0011).