

# Laboratorium Podstaw Techniki Mikroprocesorowej

Instytut Mikroelektroniki i Optoelektroniki  
Politechniki Warszawskiej



## Ćwiczenie 4

### Klawiatura i wyświetlacz ciekłokrystaliczny

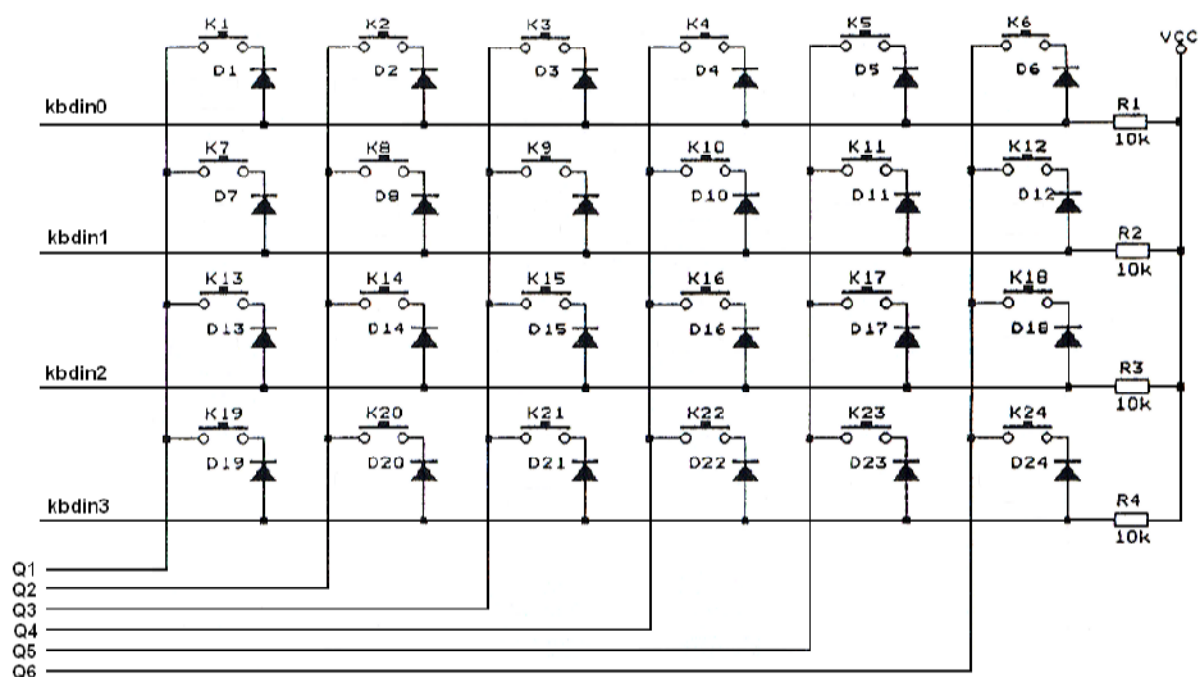
Warszawa, 2005-04-02

## 1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z programową obsługą elektromechanicznej klawiatury w układzie matrycy M kolumn x N rzędów, współpracą procedur obsługi klawiatury i wyświetlacza LCD, a także sposobem realizacji wprowadzania do programu wielkości numerycznych przy pomocy klawiatury i wyświetlacza LCD.

## 2. Opis klawiatury

Klawiatura systemu uruchomieniowego zawiera 24 przyciski i zrealizowana jest w układzie matrycowym - 6 kolumn x 4 rzędy. Schemat klawiatury przedstawiony jest na Rys. 1, natomiast schemat całego modułu zawierającego klawiaturę i wyświetlacz LCD na Rys. 5.



Rys. 1. Schemat ideowy klawiatury.

Stan klawiatury przeczytać można w procesie skanowania. Kolumny (Q1 – Q6) są wejściami, natomiast rzędy (kbdi1 – kbdi3) są wyjściami. Jednorazowo możliwe jest przeczytanie stanu jednej kolumny (4 klawiszy). Dla przeczytania stanu kolumny należy na jej wejściu (Qx) wymusić stan 0 logiczne i przeczytać stan sygnałów wyjściowych kbdi1 – kbdi3. Jeżeli żaden z klawiszy danej kolumny nie jest wciśnięty, to wszystkie sygnały wyjściowe będą miały stan 1. W przypadku, gdy któryś z klawiszy skanowanej kolumny będzie wciśnięty, to odpowiadający mu sygnał wyjściowy (kbdi1 – kbdi3) będzie miał stan 0 logiczne. Przykładowo, jeżeli skanowana jest kolumna Q1 (sygnał Q1 ma stan 0) i wciśnięty jest klawisz K7, to sygnał kbdi1 będzie miał stan 0, natomiast pozostałe sygnały będą miały stan 1.

Sygnały Q1-Q6 są sygnałami wyjściowymi rejestru U5 (Rys. 4). W celu przeczytania stanu danej kolumny należy do rejestru U5 wpisać słowo z 0 na pozycji danej kolumny, pozostałe bity muszą mieć stan 1. Sposób określania adresu rejestru U5 został podany w instrukcji do Ćwiczenia 3 – Tabela 9 i 10. Odczyt stanu skanowanej kolumny realizowany jest przez bufor U4. Sposób określania adresu bufora U4 został podany w instrukcji do Ćwiczenia 3 – Tabela 9 i 10.

Wyboru skanowanej kolumny klawiatury dokonuje się przez zapisanie odpowiedniej wartości do rejestru klawiatury U5 (w postaci binarnej liczba ta ma tylko jedno zero i 7 jedynek) np.:

```

mov     A, #11110111B      ;Q4 = 0
mov     DPTR, #adres_klaw  ;adres_klaw = adres klawiatury
movx    @DPTR, A

```

Odczytu stanu kolumny dokonujemy w sposób następujący:

```

mov     DPTR, #adres_klaw  ; adres_klaw = adres klawiatury
movx    A, @DPTR

```

Klawisze wykorzystywane w klawiaturze są elementami elektromechanicznymi, a więc charakteryzuje je pewna sprężystość. Zmianie stanu – przejścia ze stanu zwarcia do rozwarcia i na odwrót – towarzyszyć będzie zawsze stan nieustalony – drżenie styku – Rys. 2.



Rys.2. Ilustracja działania klawisza.

W procedurze obsługi klawiatury należy uwzględnić występowanie stanów nieustalonych poprzez implementację odpowiedniej procedury „antyodbiciowej”. Zdetekowaną zmianę stanu klawisza należy potwierdzić po upływie dostatecznie długiego czasu dla zakwalifikowania jej jako stabilnej. Czas trwania stanu nieustalonego  $t_{DB}$  (Rys. 2) zależy od sposobu realizacji styków i ich stanu i może być dość długi, nawet kilkadziesiąt milisekund.

Skanowanie klawiatury z procedurą antyodbiciową najwygodniej będzie zrealizować wykorzystując przerwanie czasowe licznika T0. Licznik T0 należy zaprogramować tak, aby zgłaszał przerwania co kilka-kilkanaście ms, np. co 10 ms. W kolejnych wykonywaniach procedury obsługi przerwania T0 należy skanować kolejne kolumny klawiatury. W takim przypadku każda z kolumn skanowana będzie co 60 ms (5ms x 6 kolumn). Czas ten jest dostateczny dla osiągnięcia stanu ustalonego zestyku ( $> t_{DB}$ ) po zmianie stanu. Porównanie stanu bieżącej klawiszy skanowanej kolumny ze stanem w poprzednim skanie umożliwi wykrywanie stabilnej zmiany stanu klawiszy.

### 3. Algorytm obsługi klawiszy

Program obsługi klawiatury musi wykrywać cztery stany:

- Klawisz nie był wciśnięty w poprzednim skanie a aktualnie jest wciśnięty– tę zmianę stanu trzeba potwierdzić w następnym skanie.
- Klawisz był wciśnięty w poprzednim skanie i aktualnie jest wciśnięty– stan stabilny wciśnięcia. Trzeba sprawdzić, czy kod klawisza został już wysłany na wyświetlacz (struktura danych musi umożliwiać zapis takiej informacji). Jeżeli nie, to należy

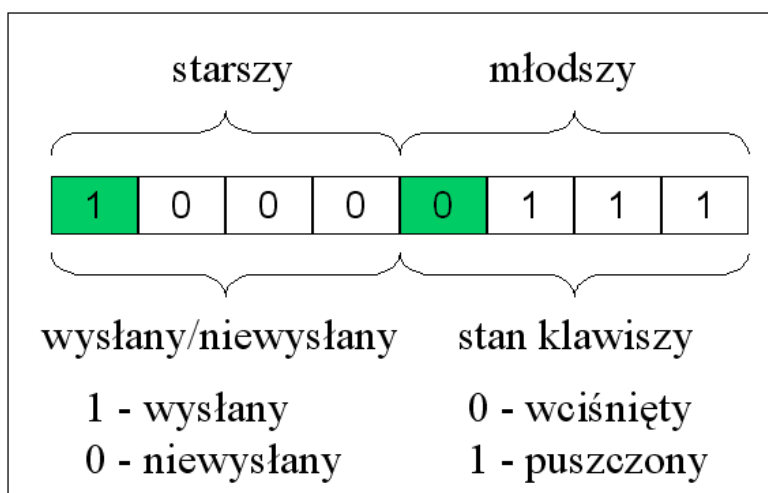
wysłać kod wciśniętego klawisza i zapisać informację o wysłaniu kodu. Jeżeli kod został wysłany poprzednio, to nie robić nic.

- c) Klawisz był wciśnięty w poprzednim skanie a aktualnie nie jest wciśnięty – tę zmianę stanu trzeba potwierdzić w następnym skanie.
- d) Klawisz nie był wciśnięty w poprzednim skanie i aktualnie nie jest wciśnięty – stan stabilny puszczenia klawisza. Należy skasować informację o wysłaniu kodu na wyświetlacz, gdyż w przeciwnym przypadku klawisz będzie "nieczynny" w przypadku ponownego wciśnięcia.

Zgodność stanu w kolejnych skanach (stany b) oraz d)) powoduje określone działanie, niezgodność (stany a) oraz c)) powoduje wyłącznie zapisanie nowego stanu klawiszy w strukturze danych klawiatury.

#### 4. Struktura danych

Do zapamiętania stanu jednej kolumny klawiatury wystarczy jeden bajt. Młodsza jego część przechowuje aktualny stan klawiszy (wciśnięty - 0, wyciśnięty - 1), odczytany w ostatnim skanie, a starsza informację o wysłaniu kodu klawisza na wyświetlacz. Na Rys. 3. przedstawiony jest przykładowy bajt pamiętający stan danej kolumny klawiatury.



Rys. 3. Bajt jednej kolumny w strukturze danych klawiatury

Zawartość bajtu na Rys. 3. wskazuje, że wciśnięty jest tylko jeden klawisz (ten, któremu odpowiada bit równy "0" w młodszej części bajtu), oraz że kod tego klawisza został już wysłany na wyświetlacz. Przyjęto tu konwencję, że "1" w starszej części bajtu oznacza, że kod klawisza został już wysłany na wyświetlacz, a "0", że klawisz nie był jeszcze „obsłużony”. Ponadto założono, że jeżeli stan danego klawisza zapisany jest np. w najstarszym bicie młodszej części bajtu (bit nr 3), to informacja o wysłaniu kodu tego klawisza na wyświetlacz jest umieszczona w najstarszym bicie starszej części bajtu (bit nr 7), itd.

Wykrycie czterech sytuacji opisanych w poprzednim punkcie następuje przez porównanie aktualnego stanu kolumny klawiatury ze stanem zapisanym w ostatnim skanie tej kolumny. Potrzeba zatem sześciu bajtów na zapisanie informacji o całej klawiaturze (6 kolumn). Struktura danych klawiatury musi zostać odpowiednio zainicjowana przed rozpoczęciem

przeglądania klawiatury: żaden klawisz nie jest wciśnięty, a na wyświetlacz nie wysłano żadnego kodu, zatem bajt odpowiadający każdej z kolumn to:

00001111B.

### 5. Komunikacja „klawiatura-wyświetlacz”

Komunikację z wyświetlaczem dobrze jest zorganizować w postaci "skrzynki pocztowej - mailbox" (jeden bajtu pamięci) oraz wskaźnika zajętości skrzynki (1 bit). W skrzynce tej procedura obsługi klawiatury przekazywała będzie kod wciśniętego klawisza do procedury obsługi wyświetlacza. Kod klawisza składa się z numeru kolumny (starsza część bajtu) i numeru klawisza w kolumnie (młodsza część bajtu). Ze skrzynką stowarzyszony jest wskaźnik zajętości (1 bit) wskazujący, czy skrzynka jest pusta (0 - mailbox\_empty), czy pełna (1 - mailbox\_full). Wskaźnik ten ustawia procedura obsługi klawiatury po wpisaniu kodu znaku do skrzynki, natomiast kasuje procedura obsługi wyświetlacza po pobraniu kodu ze skrzynki.

W przypadku złożonych programów może się zdarzyć, że procedura odbierająca znaki (tu jest to procedura obsługi wyświetlacza) nie zdąży odebrać wysłanego znaku przed wprowadzeniem z klawiatury następnego znaku. Przed wysłaniem znaku procedura obsługi klawiatury powinna więc sprawdzać, czy wysłany ostatnio znak został już odebrany. Jeżeli nie to trzeba by zorganizować przejściowe przechowywanie generowanych znaków tak, aby procedura odbierająca mogła je w stosownym czasie odebrać. Profesjonalne rozwiązanie problemu komunikacji pomiędzy procesami to zorganizowanie bufora pośredniczącego w postaci struktury danych zwanej FIFO (First-In-First-Out). Wykracza to jednak poza zakres merytoryczny Laboratorium. W programie pisanym w ramach tego Ćwiczenia zakłada się, że procedura odbierająca zawsze zdąży odebrać wysłany znak przed wprowadzeniem następnego.

### 6. Zadania do wykonania

**a) na ocenę dostateczną:** Napisać program obsługi klawiatury z wykorzystaniem przerwania czasowego licznika T0 z implementacją procedury przeciwozbiciowej. Program główny ma wysyłać kody wciśniętych klawiszy według Rys. 4. (0-9, A-F, pozostałe klawisze generują spację) na wyświetlacz LCD (kierunek przesuwania "kursora" w prawo) – w tym celu należy wykorzystać program uruchomiony w ramach Ćwiczenia 3 (wyświetlacz LCD). Po wypełnieniu pierwszej linii przejść do drugiej linii wyświetlacza. Po wypełnieniu drugiej linii wyczyścić ekran i zacząć od początku.

.....

**UWAGA!** Należy pamiętać, że do wyświetlacza LCD należy przysyłać kody ASCII znaków, które mają być wyświetlone – w programie głównym należy więc przekodować otrzymane kody klawiszy na znaki ASCII odpowiadającym im znaków.

.....

**b) na ocenę dobrą:** jak w zadaniu a) z tym, że wprowadzanie na wyświetlacz rozpoczyna się w lewym górnym rogu a po zapisaniu górnej linii następne znaki wprowadzane są na ostatnią pozycję linii (prawy górny róg) z jednoczesnym przesunięciem tekstu linii o jedną pozycję w lewo. Wprowadzenie znaku Enter przełącza linię wyświetlania wprowadzanych znaków z pierwszej na drugą, a z drugiej na pierwszą, z jednoczesnym czyszczeniem linii i rozpoczyna wprowadzanie znaków od lewej strony.

**c) na ocenę bardzo dobrą:** napisać program wprowadzania 4 stałych numerycznych (S1 – S4) w formacie  $SX = +/-x,xxx$ , np.  $S1 = -2,345$  z możliwością edycji wprowadzanej wartości (klawisz BS).

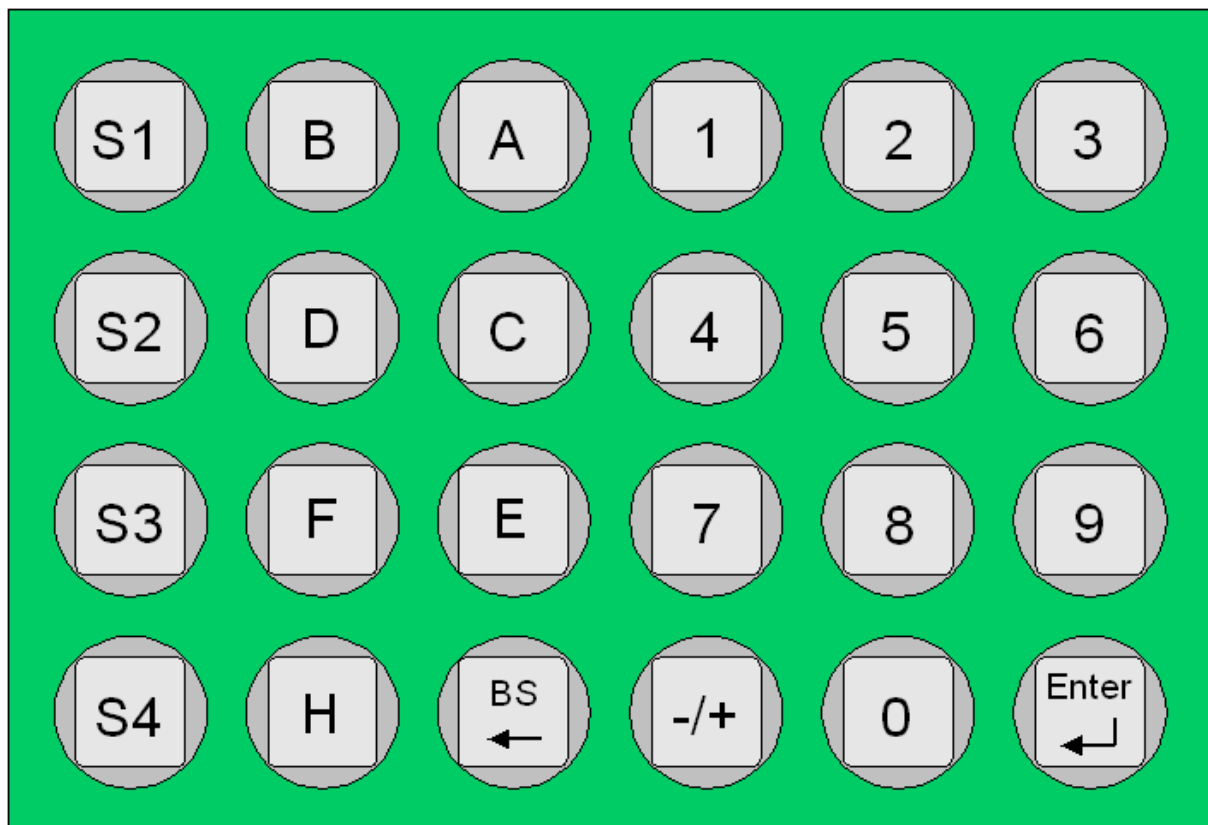
**d) na ocenę bardzo dobrą (alternatywnie):** napisać program konwersji liczb 16-bitowych z kodu heksadecymalnego  $\leftrightarrow$  kod dziesiętny.

Wciśnięcie klawisza S1 rozpoczyna wprowadzanie liczby – pojawia się komunikat S1=. Wprowadzona liczba zakończona znakiem H jest liczbą w kodzie heksadecymalnym – sprawdzenie poprawności, zamiana na kod dziesiętny i wyświetlenie wyniku. Liczba bez znaku H na końcu jest liczbą w kodzie dziesiętnym – sprawdzenie poprawności, konwersja na kod heksadecymalny i wyświetlenie wyniku.

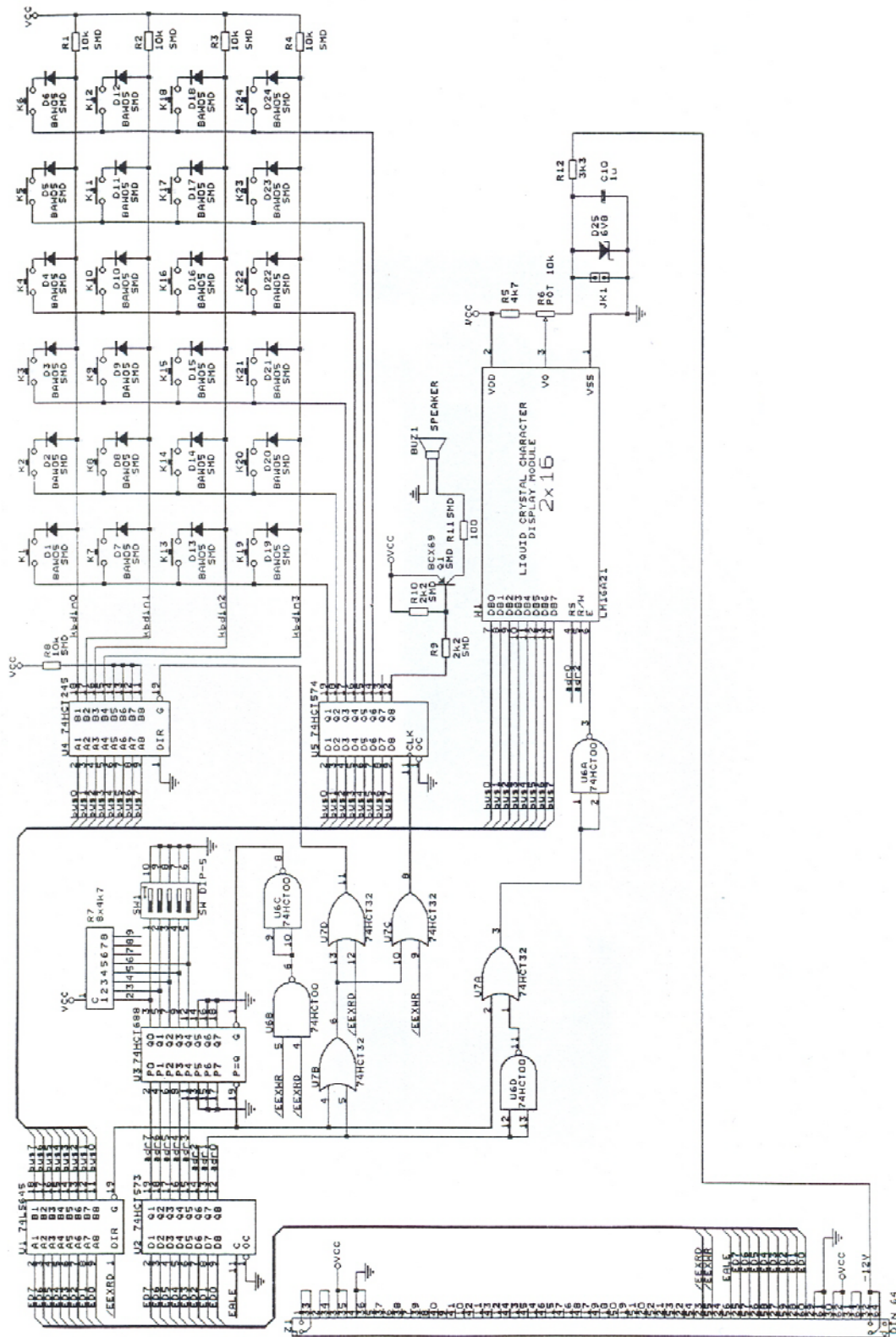
.....

**UWAGA:** Proszę pamiętać, programowanie to **algorytm + struktury danych**. Reszta to kodowanie programu. Proszę więc rozpocząć od dokładnego przemyślenia zadania i opracowania sensownego algorytmu realizacji zadania oraz odpowiedniej dla niego struktury danych.

.....



Rys.4. Układ klawiatury



Rys. 5. Schemat ideowy pakietu DBLCD