

1. Klawiatura - warstwa elektryczna.

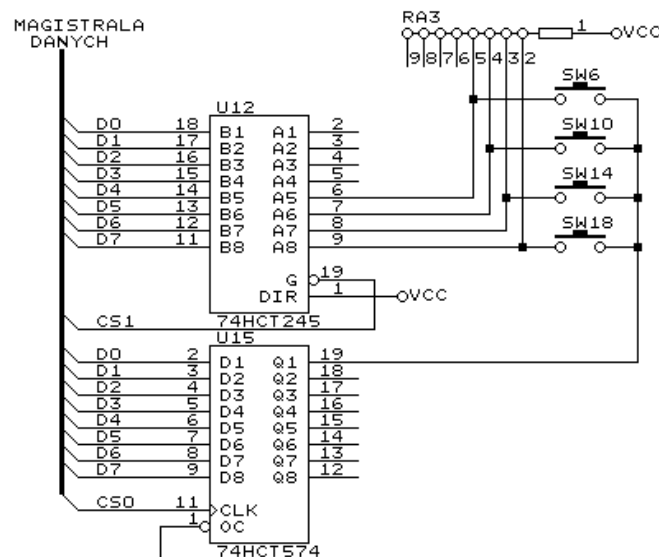
Zgodnie ze schematem zamieszczonym w opisie ćwiczenia 1, można wyróżnić dwa rodzaje klawiatury zaimplementowane w systemie STRC51:

-Pierwsza (klawisze SW2, SW3, SW4, SW5) podłączona do U12.

Naciśnięcie klawiszy powoduje podanie na wejścia A1, A2, A3, A4 układu U12 stanu odpowiadającego logicznemu „0”, przy zwolnionych klawiszach podawane są odpowiednio stany logicznej „1”.

-Druga (klawisze od SW6 do SW21), zbudowana w oparciu o układy U15 i U12.

Dla dalszej analizy działania tej klawiatury zastosowany zostaną oznaczenia: kolumny, wiersze. Kolumny podłączone są do U15 a wiersze do U12. Podczas pracy z tą klawiaturą należy podać odpowiednio stany na wyjścia U15, czytając z U12 stany można ustalić który klawisz został wciśnięty. Podanie na odpowiednie wyjście U15 stanu „0” (czyli kolumnę), spowoduje że ewentualne zwarcie klawisza doprowadzi do przeniesienia „0” na odpowiednie wejście U12. O tym na które wejście (wiersz) decyduje pozycja klawisza. Rys. 1 pokazuje przykład pracy.



Rys.1 Schemat poglądowy pracy układu klawiatury

Zastosowany rezystor podciągający (miedzy A8 a VCC na rysunku 1.) sprawia, że w stanie spoczynkowym (klawisze zwolnione) na wejścia U12 podawane są „1”. Zatem w stanie gdy żaden klawisz nie jest naciśnięty, bez względu na stan wyjść U15 na wejściach U12 pojawi się stan „1”. Podczas programowania U15 nie należy zapomnieć o drugiej funkcji tego układu – sterowaniu wyświetlacza LED.

2. Klawiatura – warstwa programowa.

Zgodnie z informacjami podanymi w poprzednich ćwiczeniach, deklaracje U12 i U15 są odpowiednio:

```
xdata at 0x8000 unsigned char U15;
xdata at 0x8000 unsigned char U12;
```

Uwagę należy zwrócić na niemalże identyczny zapis obu deklaracji. Różnica polega na funkcji U15 i U12. Deklaracja dla układu U15 służy tylko do zapisu a U12 tylko do odczytu.

Obsługa programowa klawiatury sprowadza się do odpowiedniego sterowania wyjściami U15 i badaniu stanu U12.

Temat: Obsługa klawiatury w systemie STRC51. Ćwiczenie 4.

Jedną z możliwych implementacji jest utworzenie takiego rozwiązania w którym w przerwaniu generowanym przez przeładowany jeden z liczników, obsługiwany był wyświetlacz LED i klawiatura. Rozwiązanie takie wydaje się być naturalnym ponieważ U15 jest współdzielony przez klawiaturę i wyświetlacz.

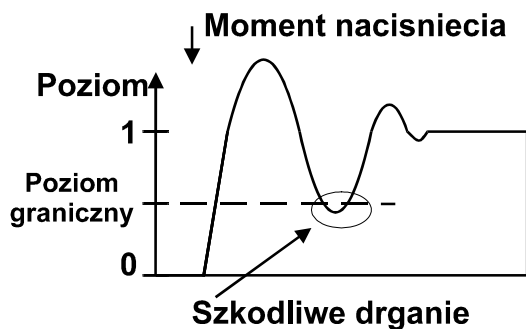
W najprostszym przypadku wystarczy „przemiatać” stanem „0” kolumny i sprawdzać stan wierszy. Funkcja `getchar()` zwróci kombinację wiersza i kolumny będącą numerem klawisza. Lepszym i bardziej zgodnym z ogólnie przyjętym działaniem funkcji jest implementacja w której czeka się na naciśnięcie klawisza, czyli „przemiatamy” aż do napotkania sytuacji w której na wejściu U12 pojawi się na co najmniej jednym wierszu stan „0”. Przemiatanie stanem „0” jest procesem podawania na U15 (odpowiednie wyjścia) logicznego „0” tylko na jedno wyjście w danym czasie, poczym podanie tego stanu na inne wyjście i tak w koło.

Innymi problemami są wielokrotnie naciśnięte klawisze i eliminacja drgań styków.

Pierwszy problem wymaga arbitralnego wybrania algorytmu odrzucającego obsługę sytuacji wielokrotnych naciśnień lub buforowania powtórzeń:

- Odrzucanie wielokrotnych naciśnień może być obsługiwane w taki sposób, gdzie tylko pierwszy bądź ostatni klawisz będzie zapamiętywany.
- Obsługa z buforowaniem powtórzeń daje informacje w kolejnych wywołaniach `getchar()`, jakie klawisze były równocześnie naciśnięte.

Drgań zestyków są związane z zjawiskami fizycznymi. Rys. 2 pokazuje wykres czasowy napięcia podczas naciskania klawisza – polaryzacja sygnału nie została zachowana w zgodzie z systemem STRC51.



Rys. 2. Proces drgania zestyków

Jedną z metod eliminacji drgań jest wielokrotne badanie klawisza i na tej podstawie wypracowywanie wyniku: naciśnięto klawisz bądź nie.

Temat: Obsługa klawiatury w systemie STRC51. Ćwiczenie 4.

3. Zadania do wykonania.

W ramach zajęć, należy zaimplementować następujące funkcje:

<code>void initkey(void);</code>	- funkcja inicjacji automatu obsługi klawiatury,
<code>unsigned char getchar(void);</code>	- funkcja zwracająca numer klawisza „ostatnio” naciśniętego, jeżeli „ostatnio” nie naciśnięto klawisza – funkcja czeka na takie zdarzenie,
<code>unsigned char ischar(void);</code>	- funkcja zwraca informacje o zdarzeniu naciśnięcia klawisza, pozwala ona na obsługę bez blokowania testowania stanu klawiatury.

W razie uzasadnionych powodów można dobrać inny zestaw funkcji, jak i typów zmiennych przekazywanych do funkcji.

4. Zasady oceniania:

- 5 - Implementacja współdziałająca z wyświetlaczem LED, wykorzystująca system przerwań, nienaganna od strony zużycia zasobów i sposobu zakodowania.
- 4 - Implementacja współdziałająca z wyświetlaczem LED.
- 3 - minimalna (najprostsza) w pełni działająca implementacja funkcji z pkt. 3.

Uwaga: należy zapewnić w jakiś sposób wizualizację działania systemu.

Prowadzący zastrzega sobie możliwość innego oceniania w przypadkach szczególnych. Mile widziane są programy poprawnie skomentowane i sformatowane (może wpływać na ocenę).

UWAGA!!! Ostrzega się, że w przypadku wątpliwości co do autorstwa kodu możliwe jest poddanie gruntownemu przeegzaminowaniu z poznanego materiału, nie wyłączając obniżenia oceny z danego ćwiczenia lub całego przedmiotu. W przypadkach skrajnych sprawa może być przekazana władzom uczelni.