Programowanie mikrokontrolerów

Klawiatury

Marcin Engel Marcin Peczarski

Instytut Informatyki Uniwersytetu Warszawskiego

6 listopada 2012

Klawiatura matrycowa

- Gdy mamy do obsłużenia więcej klawiszy, to możemy każdy podłączyć do innego wyprowadzenia.
- Przy dużej liczbie klawiszy zabraknie portów w mikrokontrolerze.
- Dużo efektywniejszym sposobem jest połączenie klawiszy w matrycę.
- Tak skonstruowana jest np. klawiatura w PC.

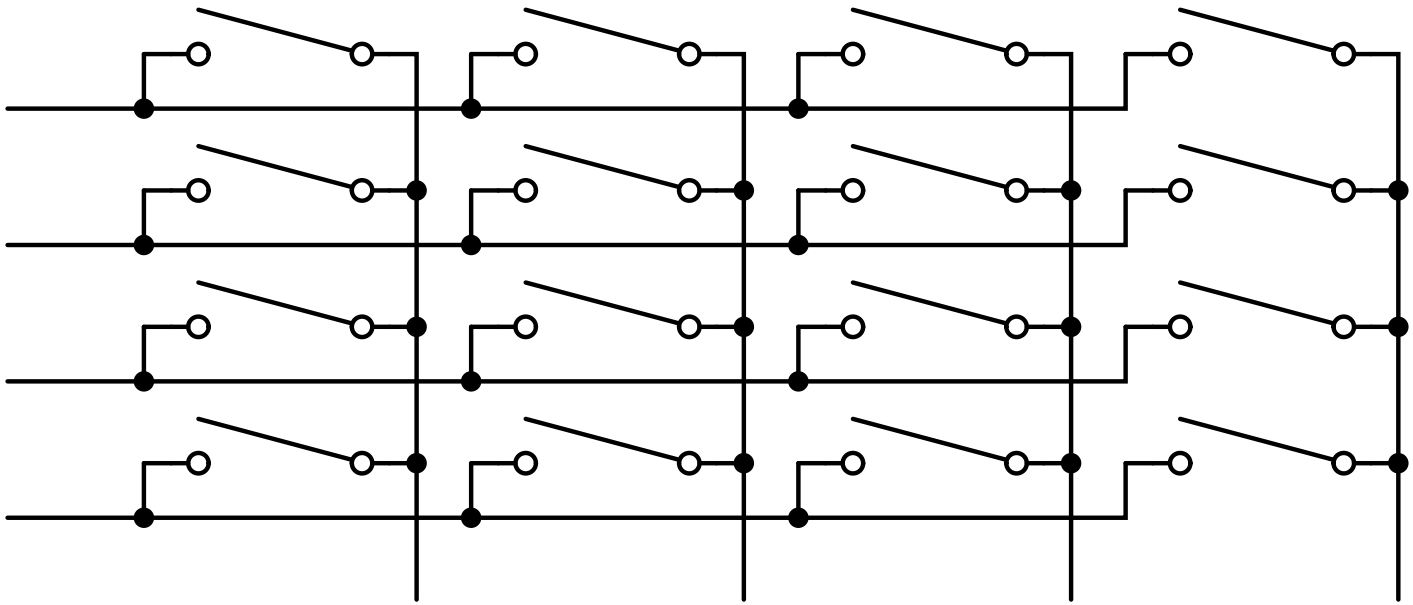
Klawiatura matrycowa 16-klawiszowa

- Bardzo popularnym typem klawiatury matrycowej jest klawiatura 4×4.

- 16 klawiszy ułożonych jest w 4 wiersze i 4 kolumny.
- Można ją spotkać np. w bankomatach, czytnikach kodów otwierających drzwi (domofony) itp.

Klawiatura matrycowa 16-klawiszowa, cd.

- Naciśnięcie klawisza zwiera linię wiersza z linią kolumny.



- W mikrokontrolerze potrzebujemy po jednym wyprowadzeniu na każdy wiersz i na każdą kolumnę.
- Do podłączenia wystarczy jeden port mikrokontrolera.

Klawiatura matrycowa w VMLAB

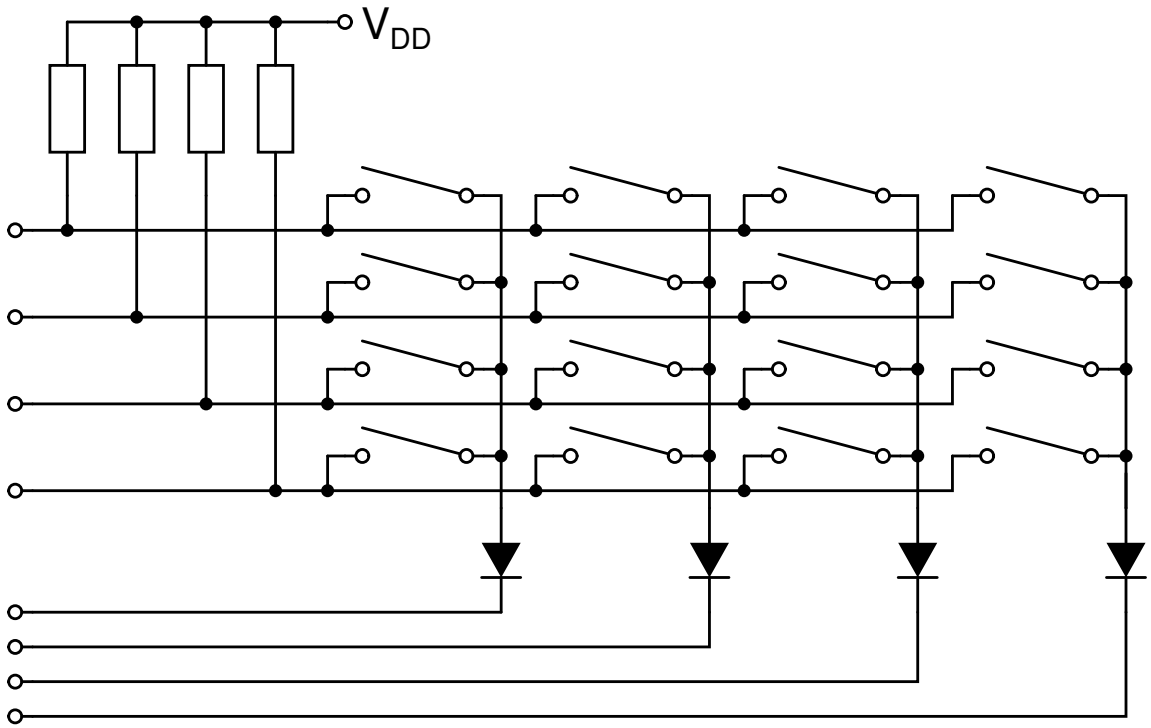
- VMLAB umożliwia symulowanie klawiatury matrycowej 4×4.
- Przyciski w panelu kontrolnym ustawione są w 4 rzędy po 4 kolumny.

```
Xnazwa KEY4x4 r1 r2 r3 r4 c1 c2 c3 c4
```

- `nazwa` jest dowolnym unikatowym identyfikatorem – bez znaczenia.
- Jako `r1`, ..., `c4` należy wpisać oznaczenia wyprowadzeń mikrokontrolera podłączonych odpowiednio do wierszy i kolumn.

Skanowanie klawiatury matrycowej

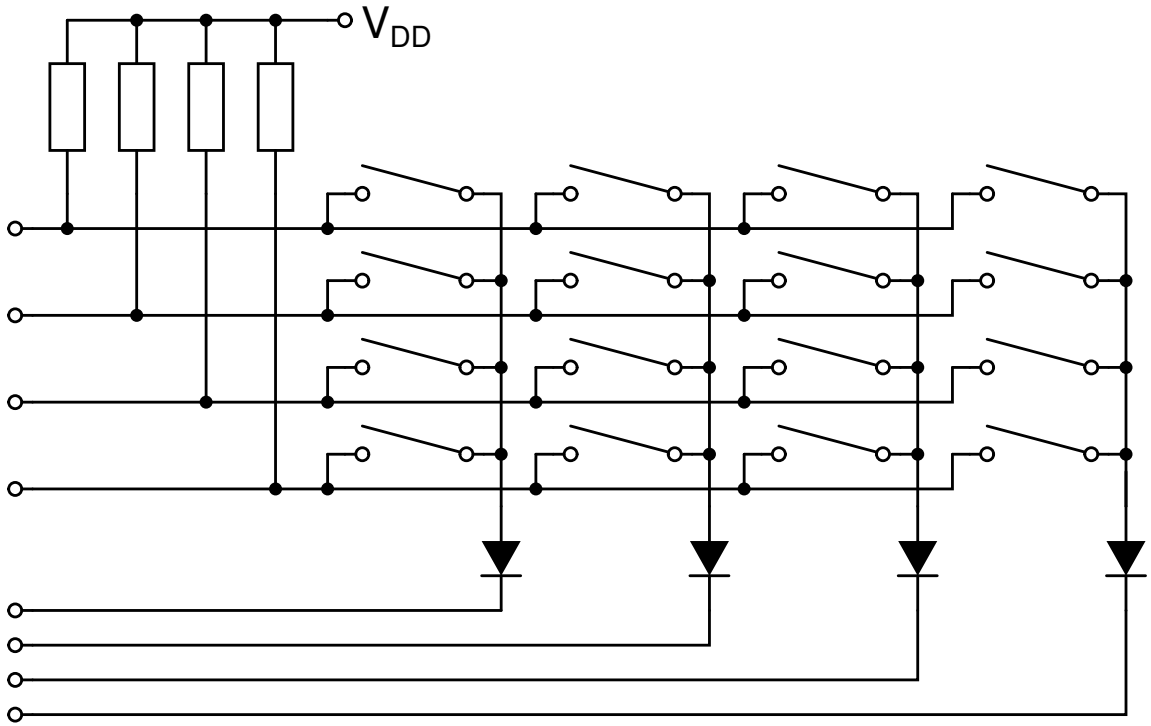
- Linie wierszy pracują jako wejścia.
- Na liniach wierszy ustalamy stan wysoki za pomocą rezystorów podciągających (mogą być wewnętrzne).
- Linie kolumn pracują jako wyjścia.
- Na liniach kolumn ustalamy stan wysoki.



- Pojęcia wierszy i kolumn są umowne, można je zamienić.

Skanowanie klawiatury matrycowej, cd.

- Cyklicznie zmieniamy stan jednej z linii kolumn na niski.
- Jeśli wciśnięty zostanie klawisz, to stan niski z jego kolumny przenosi się na jego wiersz.
- Prąd płynie od plusa zasilania (VDD) przez rezystor podciągający, styki klawisza, diodę, wyjście kolumny do masy.



- Diody zabezpieczają przed zwarciem linii kolumn, gdy wciśnie się równocześnie klawisze w różnych kolumnach.

Algorytm skanowania klawiatury matrycowej

- Inicjacja

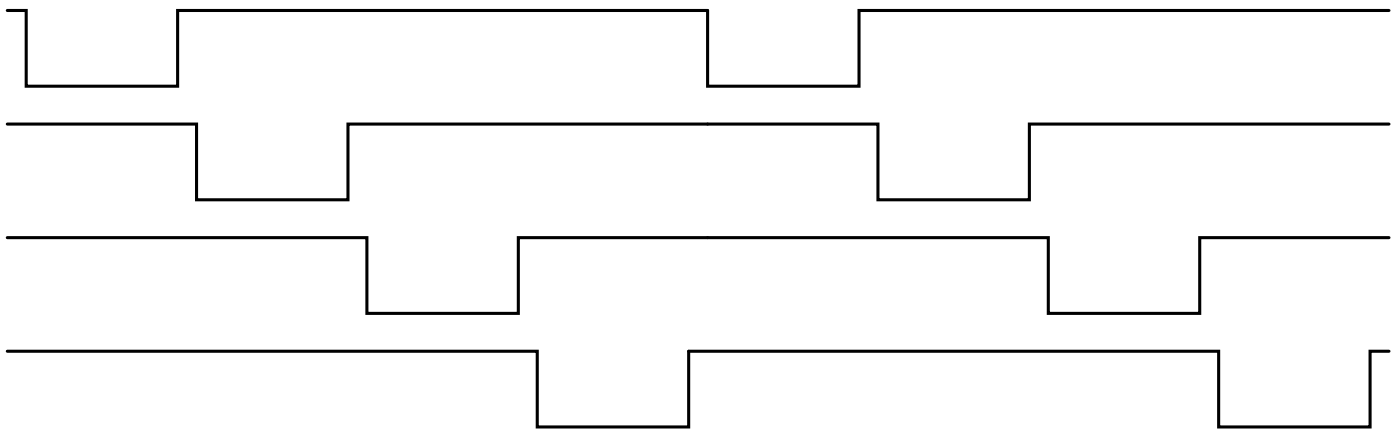
```
ustaw linie WIERSZY jako WEJŚCIA;
ustaw REZYSTORY podciągające na liniach WIERSZY;
ustaw linie KOLUMN jako WYJŚCIA;
ustaw stan WYSOKI na wszystkich liniach KOLUMN;
```

- Właściwe skanowanie

```
for (kolumna := 1 to liczba_kolumn) do
begin
  ustaw stan niski na linii kolumna;
  wait(T0);
  for (wiersz := 1 to liczba_wierszy) do
    if (linia wiersz w stanie niskim) then
      obsłuż zdarzenie klawisz(kolumna, wiersz);
  ustaw stan wysoki na linii kolumna
end
```

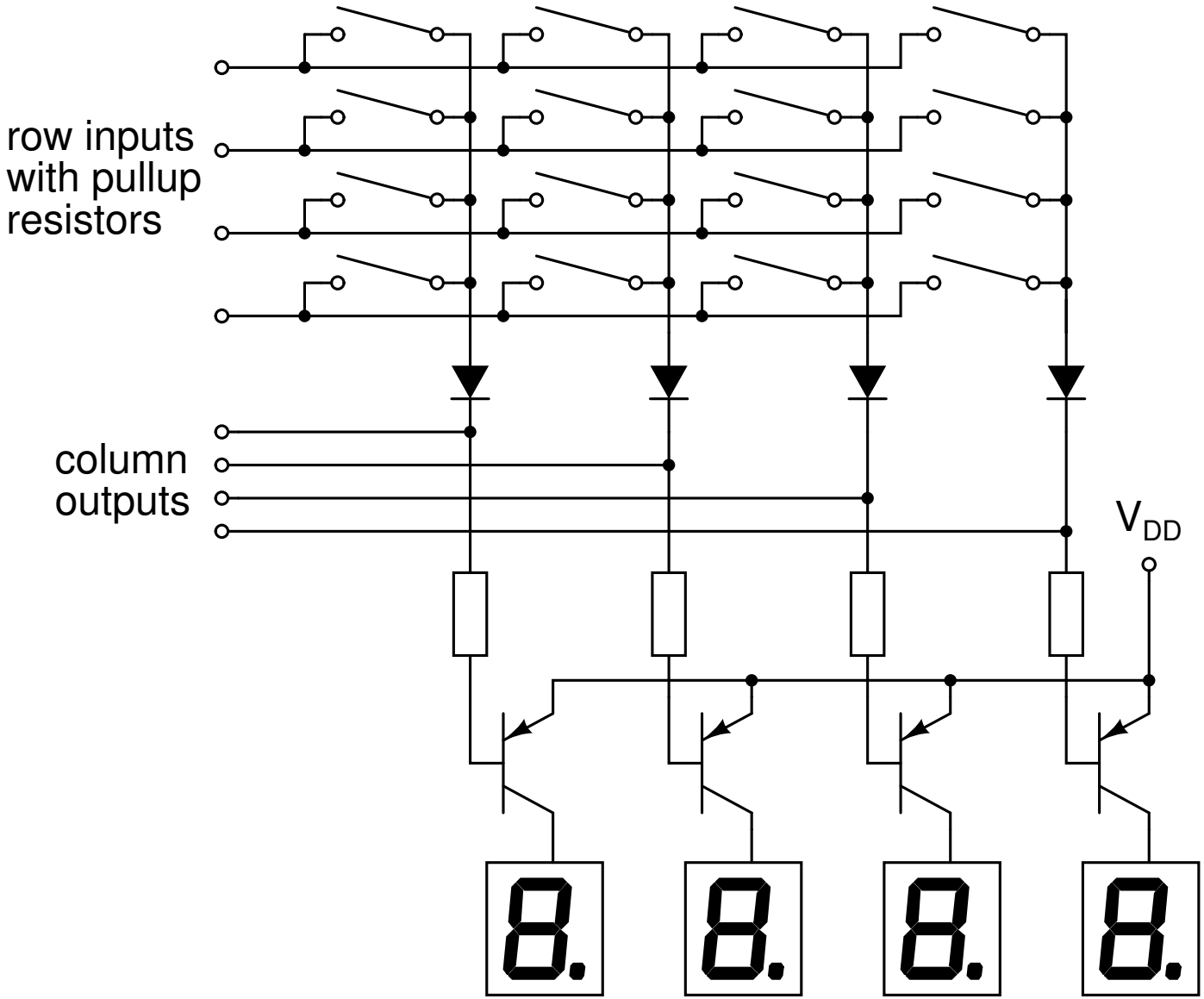
Przebiegi sygnałów na liniach kolumn

- Jak wygenerować takie przebiegi bez nadmiernego wysiłku?
- Patrz wykład o licznikach.



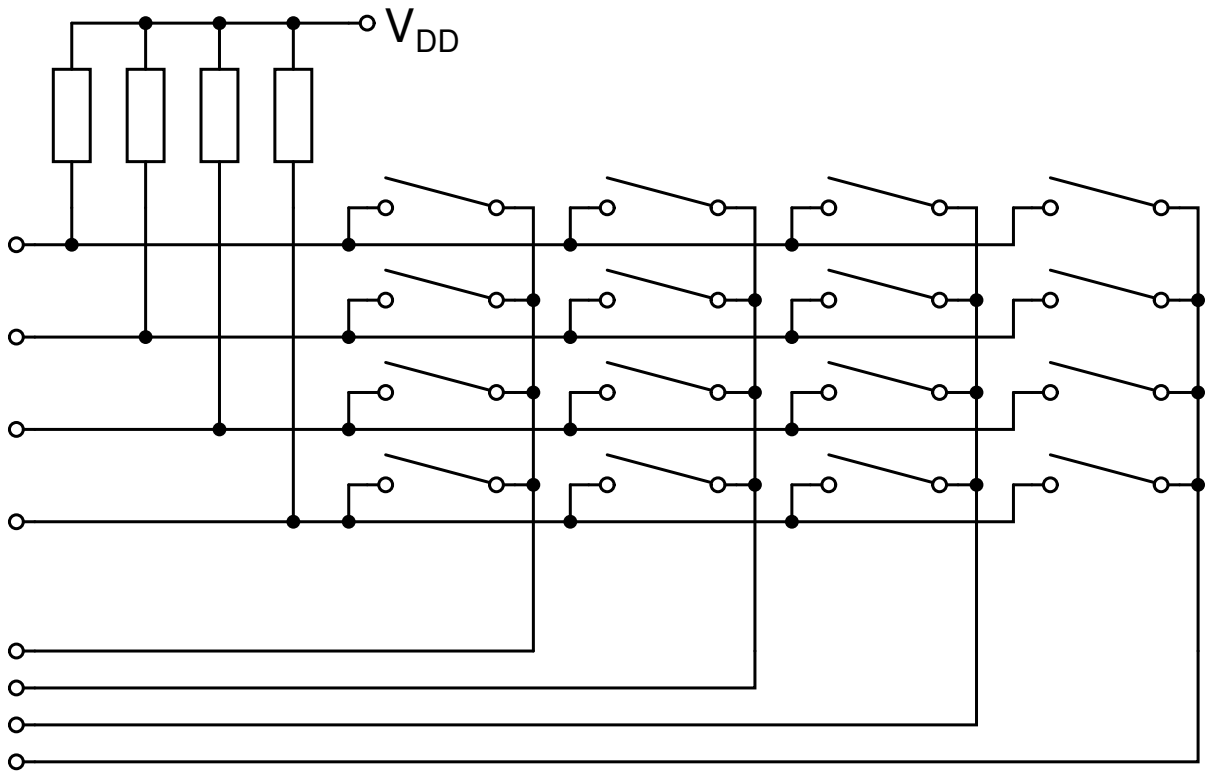
- Jak już mamy takie przebiegi, to możemy nimi sterować równocześnie klawiaturę i zestaw wyświetlaczy 7-segmentowych...

Wspólne multipleksowanie



Szybkie skanowanie klawiatury matrycowej

- Firma Atmel zaleca inny sposób skanowania klawiatury 4×4.



- Nie są potrzebne żadne dodatkowe elementy zewnętrzne:
  - rezystory podciągające mogą być wewnętrzne,
  - diody też są zbędne.

Algorytm szybkiego skanowania klawiatury matrycowej

- Inicjacja

```
ustaw linie WIERSZY jako WEJŚCIA;
ustaw rezystory podciągające na liniach WIERSZY;
ustaw linie KOLUMN jako WYJŚCIA;
ustaw stan NISKI na wszystkich liniach KOLUMN;
```

- Właściwe skanowanie

```
l_wierszy := 0;
for (wiersz := 1 to liczba_wierszy) do
  if (linia wiersz w stanie niskim) then
  begin
    l_wierszy := l_wierszy + 1;
    w_wiersz := wiersz
  end
end
```

Algorytm szybkiego skanowania klawiatury matrycowej, cd.

- Skanowanie, cd.

```
ustaw linie KOLUMN jako WEJŚCIA;
ustaw rezystory podciągające na liniach KOLUMN;
ustaw linie WIERSZY jako WYJŚCIA;
ustaw stan NISKI na wszystkich liniach WIERSZY;

wait(T0);
l_kolumn := 0;
for (kolumna := 1 to liczba_kolumn) do
begin
  if (linia kolumny w stanie niskim) then
  begin
    l_kolumn := l_kolumn + 1;
    w_kolumna := kolumna
  end
end
```

Algorytm szybkiego skanowania klawiatury matrycowej, cd.

- Dokończenie skanowania

```
ustaw linie WIERSZY jako WEJŚCIA;
ustaw rezystory podciągające na liniach WIERSZY;
ustaw linie KOLUMN jako WYJŚCIA;
ustaw stan NISKI na wszystkich liniach KOLUMN;

if (l_wierszy = 1 and l_kolumn = 1) then
  obsłuż zdarzenie klawisz(w_kolumna, w_wiersz);
```

Przykładowa implementacja (1)

- Wbrew pozorom dla klawiatury 4×4 ten algorytm można zaimplementować bardzo sprytnie bez użycia pętli.
- Dla ustalenia uwagi przyjmijmy, że klawiatura jest podłączona do portu PB i port został właściwie zainicjowany.
- Zaczynamy od testowania linii wierszy.

```
ldi     r17, -58
sbis    PINB, PB0
subi    r17, -41
sbis    PINB, PB1
subi    r17, -45
sbis    PINB, PB2
subi    r17, -49
sbis    PINB, PB3
subi    r17, -53
```

Przykładowa implementacja (2)

- Zmieniamy kierunki portu klawiatury: linie wierszy 0–3 jako wyjścia, linie kolumn 4–7 jako wejścia.

```
ldi     r16, 0x0F
out     DDRB, r16
```

- Ustawiamy stan niski na liniach wierszy i podciąganie na liniach kolumn.

```
ldi     r16, 0xF0
out     PORTB, r16
```

- Czekamy ok. 1 μs na ustalenie się poziomów. W zależności od częstotliwości zegara od 1 do 16 rozkazów `nop`.

```
nop
nop
nop
nop
```

Przykładowa implementacja (3)

- Testujemy linie kolumn.

```
sbis    PINB, PB4
subi    r17, -17
sbis    PINB, PB5
subi    r17, -18
sbis    PINB, PB6
subi    r17, -19
sbis    PINB, PB7
subi    r17, -20
```

- Przywracamy pierwotną konfigurację klawiatury: linie wierszy 0–3 jako wejścia i linie kolumn 4–7 jako wyjścia.

```
out     DDRB, r16   ; r16 == 0xF0
```

- Ustawiamy podciąganie na liniach wierszy i stan niski na liniach kolumn.

```
ldi     r16, 0x0F
out     PORTB, r16
```

Przykładowa implementacja (4)

- Sprawdzamy, czy udało się poprawnie zeskanować?

```
cpi     r17, 16
brsh    NO_KEY_PRESSED
```

- Jeśli doszliśmy do tego miejsca, to `r17` zawiera numer wciśniętego klawisza, z przedziału 0 do 15.

Podsumowanie

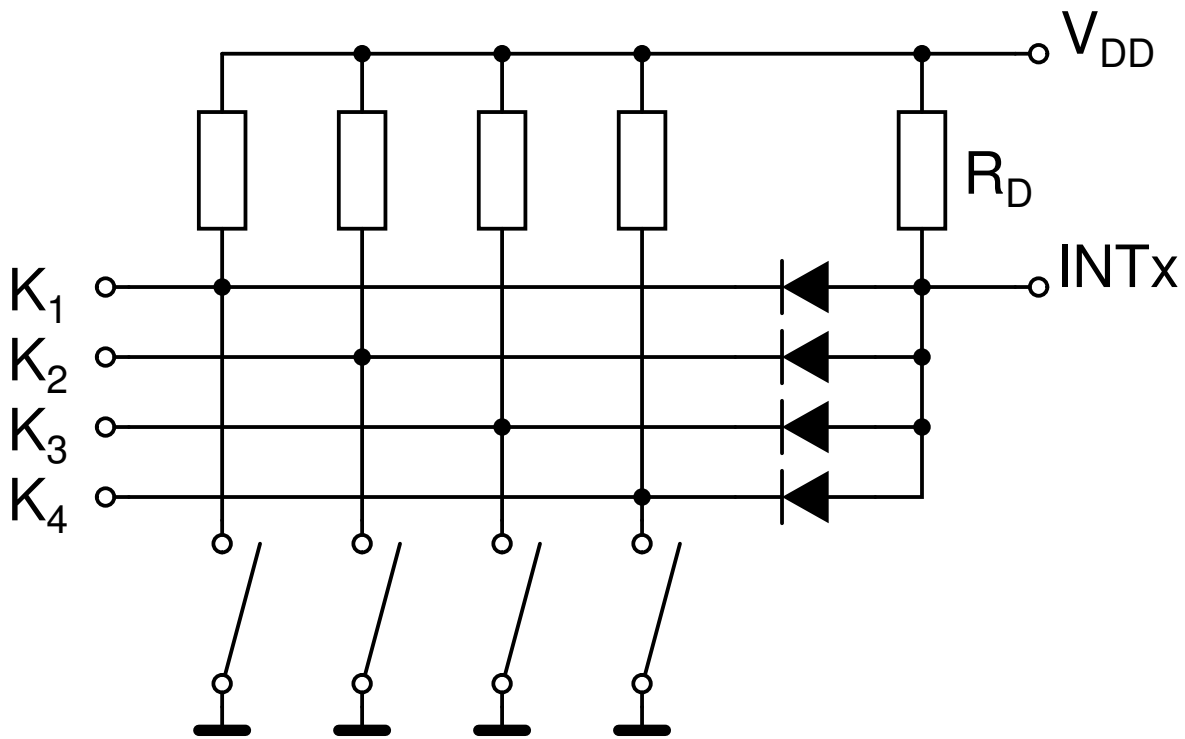
- Pokazaliśmy, jak mikrokontroler może aktywnie czytać stan klawiszy.
- Czasem nie chcemy lub nie możemy marnować cykli zegara na aktywne oczekiwanie – potrzebujemy wykonywać inne obliczenia.
- Rozwiązaniem są przerwania, ale o nich za chwilę...
- Oczywiście styki klawiatury matrycowej też mogą drgać.
- Często też chcemy, aby w klawiaturze matrycowej było autopowtarzanie.
- Trzeba więc połączyć algorytm dla jednego klawisza z algorytmem skanowania.
- Miłej zabawy...

Wykorzystanie przerwań do podłączenia przycisków

- Jeden przycisk podłączony bezpośrednio do wyprowadzenia INT0, INT1 lub INT2:
  - wciśnięcie powoduje pojawienie się przerwania zewnętrznego;
  - nie trzeba aktywnie sprawdzać stanu przycisku;
  - o obsłudze przerwań było w zeszłym tygodniu.
- A co, gdy chcemy mieć więcej niż 3 klawisze?
  - Niektóre nowsze mikrokontrolery AVR mają więcej źródeł przerwań zewnętrznych, np. zmiana poziomu na dowolnym wyprowadzeniu portu.
  - ATmega16 ma ich niestety tylko 3.
- Rozwiązanie:
  - podłączenie kilku przycisków do osobnych wyprowadzeń;
  - dodatkowa linia sygnalizująca wciśnięcie dowolnego z nich podpięta do INT0, INT1 lub INT2;
  - wciśnięcie dowolnego przycisku powoduje pojawienie się przerwania zewnętrznego;
  - procedura obsługi przerwania wykrywa, który przycisk został wciśnięty.
- Potrzebna jest bramka logiczna OR...

Cztery klawisze

- Diody wraz z rezystorem podciągającym R_D tworzą bramkę OR pracującą w logice ujemnej:
  - wciśnięcie przycisku to stan niski;
  - na wyprowadzeniu INTx pojawia się stan niski, gdy co najmniej jeden przycisk jest wciśnięty.



- Rezystory podciągające mogą być jak zwykle wewnętrzne.

Cztery klawisze, cd.

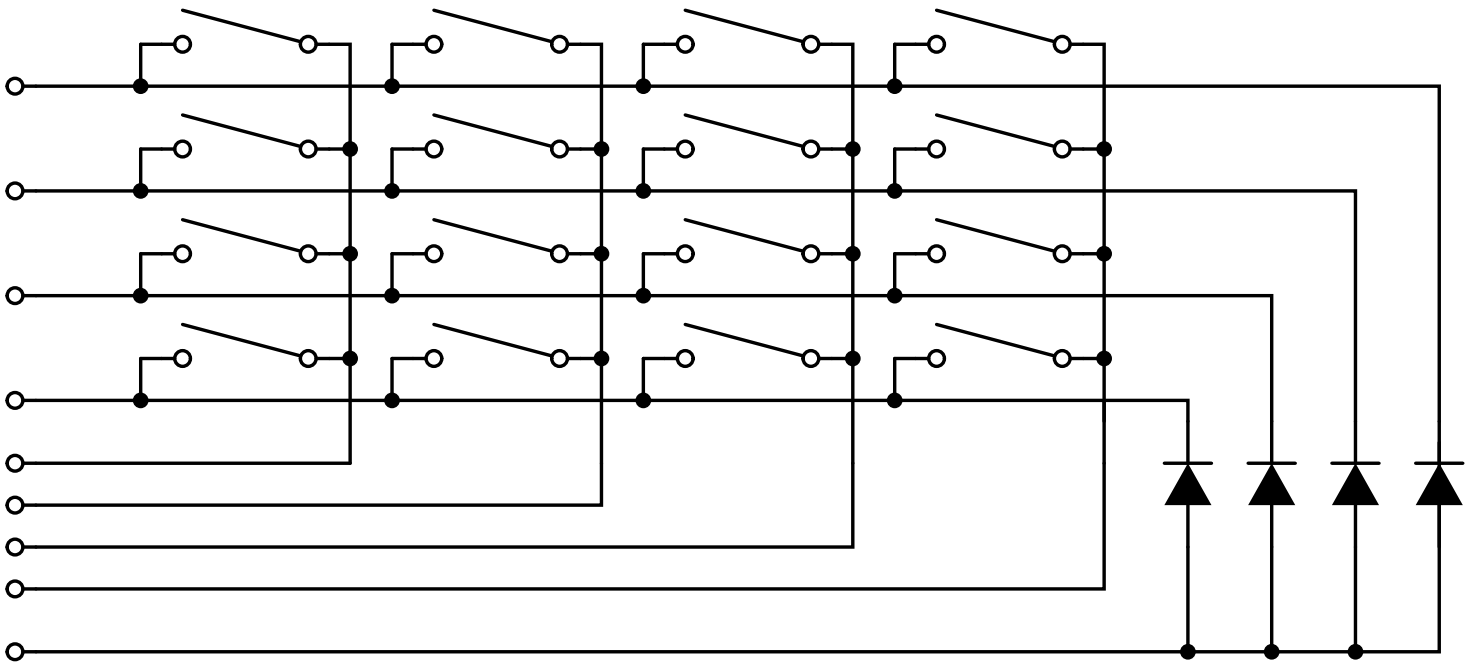
- Układ z poprzedniego rysunku można w VMLAB zasymulować, używając bramek NAND:

```
K1 GND PA0
K2 GND PA1
K3 GND PA2
K4 GND PA3
X ND2 PA0 PA1 N1
X ND2 PA2 PA3 N2
X ND2 N1 VDD N3
X ND2 N2 VDD N4
X ND2 N3 N4 N5
X ND2 N5 VDD PD2
```

- W powyższym klawisze `K1`, . . . , `K4` podłączone są odpowiednio do wyprowadzeń `PA0`, . . . , `PA3` i użyte jest wejście przerwania INT0 (wyprowadzenie `PD2`).

Klawiatura matrycowa z wykorzystaniem przerwania

- Jeśli stosujemy klawiaturę 4×4 z szybkim skanowaniem, to odpowiedni schemat wygląda tak (nie uwzględniono wewnętrznych rezystorów podciągających):



- Przypomnijmy, że w stanie między skanowaniami na liniach kolumn jest stan niski.
- Wciśnięcie dowolnego przycisku spowoduje więc zmianę stanu na wyprowadzeniu przerwania na niski i zgłoszenie przerwania.

Klawiatura matrycowa z wykorzystaniem przerwania, cd.

- Procedura obsługi przerwania wykonuje skanowanie klawiatury.
- Podczas procedury skanowania manipulujemy poziomami na liniach wierszy i może zostać zgłoszone fałszywe przerwanie.
- Należy pamiętać, aby je skasować po zakończeniu procedury skanowania, a przed zakończeniem obsługi przerwania.
- Układ diod można w VMLAB zasymulować jak poprzednio, używając bramek NAND.