

## 1. Zagadnienia

- Linie I/O, porty wejścia/wyjścia
- Rejestry związane z pracą portów
- Praca linii I/O jako wyjścia - ograniczenia prądowe
- Symulacje w programie SimulIDE

## 2. Porty wyjściowe/wyjściowe

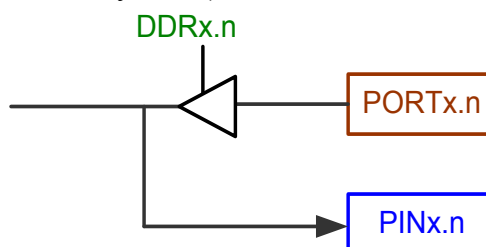
W mikrokontrolerze ATmega16 mamy do dyspozycji 32 linie I/O (wejścia/wyjścia) grupowane po 8, nazywane portami. Oznaczone są kolejno literami A, B, C, D. Każda linia portu może niezależnie pracować jako wejście lub jako wyjście cyfrowe. Dodatkowo linie mogą mieć różne alternatywne zastosowania (np. wejścia analogowe przetwornika A/C, linie przerwań, linie interfejsów szeregowych, itp.)

Podstawowe zastosowanie linii portów jako we/wy cyfrowych wymaga obsługi trzech rejestrów (w sposób uproszczony przedstawiono to na rysunku):

DDRx - rejestr kierunkowy

PORTx - rejestr wyjściowy

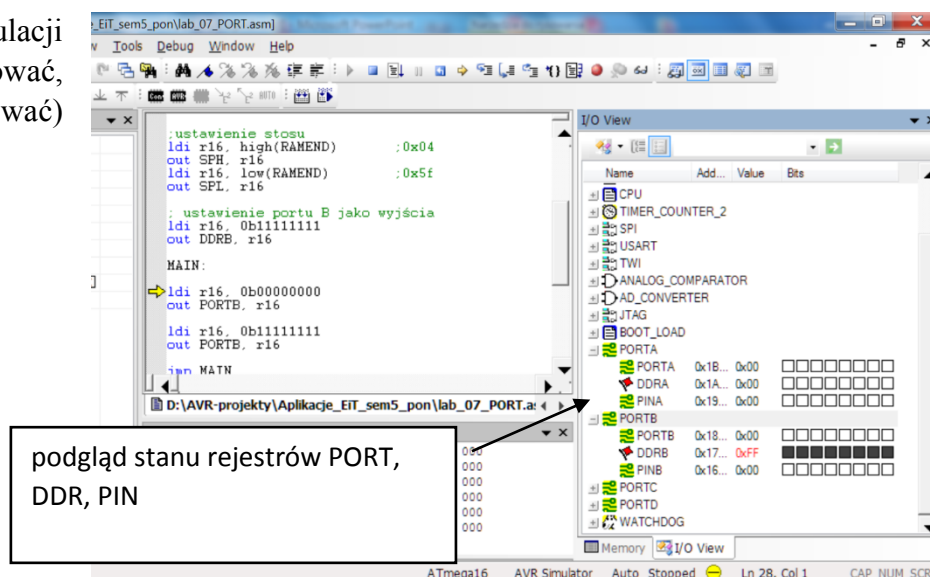
PINx - rejestr wejściowy



Rejestr kierunkowy DDRx określa kierunek linii. Wpisanie 1 ustawia linię jako wyjściową, wpisanie 0 - ustawia linię jako wejściową. Wysyłanie danych odbywa się poprzez rejestr PORTx, odczyt danych odbywa się z wykorzystaniem rejestru PINx (x - oznacza odpowiednio port A, B, C, D). Rejestry DDRx, PORTx, PINx znajdują się w obszarze rejestrów specjalnych, do zapisu i odczytu tych rejestrów wykorzystujemy instrukcje OUT i IN.

## 3. Podgląd stanu portów podczas symulacji

W trakcie symulacji możemy obserwować, (a także modyfikować) stan portów.



## 4. Praca portu jako wyjścia

Napisz program ustawiający port B jako wyjście. Zmieniaj cyklicznie stan wyjścia z 1 na 0:

Program\_01

```
.nolist
#include "m16def.inc"
.list
.listmac
.device ATmega16

.cseg
.org 0x0000
jmp START

;wektory przerwań

.org 0x0030
START:

;ustawienie stosu
ldi r16, high(RAMEND)      ;0x04
out SPH, r16
ldi r16, low(RAMEND)       ;0x5f
out SPL, r16

;ustawienie portu B jako wyjścia
ldi r16, 0b11111111
out DDRB, r16

MAIN:

ldi r16, 0b00000000
out PORTB, r16

ldi r16, 0b11111111
out PORTB, r16

jmp MAIN
```

ustawienie  
wszystkich linii portu  
B jako wyjściowych

Wysłanie wartości na  
wycie portu B

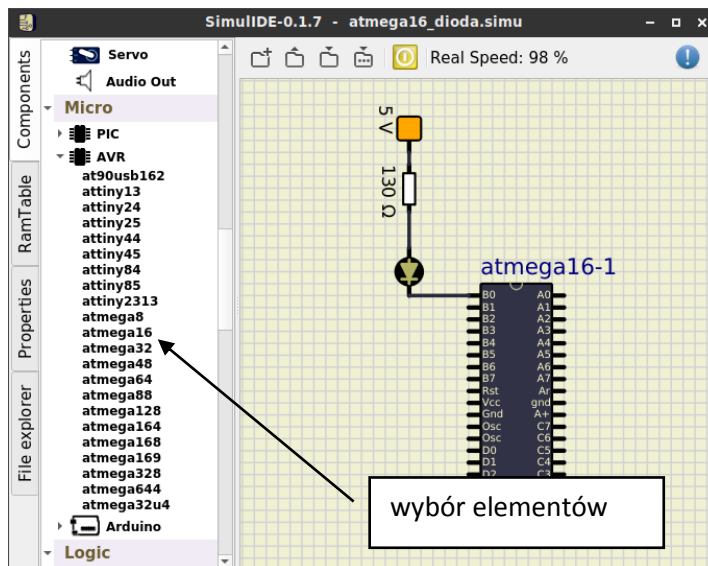
Program cyklicznie zmienia stan portu B z 1 na 0. Dlaczego uzyskany przebieg nie ma symetrycznego wypełnienia ("1" trwa dłużej niż "0"). Zastosuj instrukcję NOP, aby uzyskać symetryczne wypełnienie "1" i "0". Podczas symulacji "zmiar" jaką częstotliwość fali prostokątnej uzyskujemy na wyjściu (obserwuj jedną linię, np. PB0). Zakładamy domyślną częstotliwość zegara 4MHz.

## 5. Symulacja działania procesora w programie SimulIDE.

Zainstaluj program SimulIDE (<https://www.simulide.com>). Narysuj schemat zawierający mikroprocesor ATmega16 z podłączoną diodą LED do linii PB0 portu B. Wstaw napięcie zasilania 5V i masę GND.

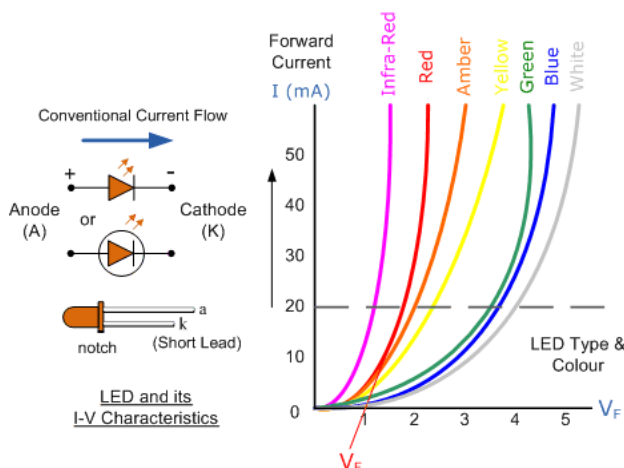
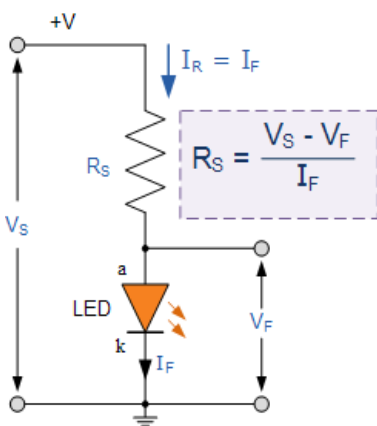
Wprowadź odpowiednią wartość rezystancji rezystora (**Prawy-Przycisk-Myszy > Properties > Resistance**). Ustaw napięcie przewodzenia diody LED dla wybranego koloru (**Prawy-Przycisk-Myszy > Properties > Treshold**).

Zauważ, że takie podłączenie diody LED wymaga podania "0" logicznego do "zaświecenia" diody. Oblicz wartość rezystora tak, aby przez diodę płynął prąd 20mA (choć nowoczesne super-jasne diody LED mogą pracować poprawnie z prądem nawet 1mA).



Przykładowo, dla napięcia zasilania  $V_S=5V$  i diody LED w kolorze żółtym  $V_F=2,4V$  otrzymujemy:

$$R_S = \frac{V_S - V_F}{I_F} = \frac{5V - 2,4V}{20mA} = 130\Omega$$

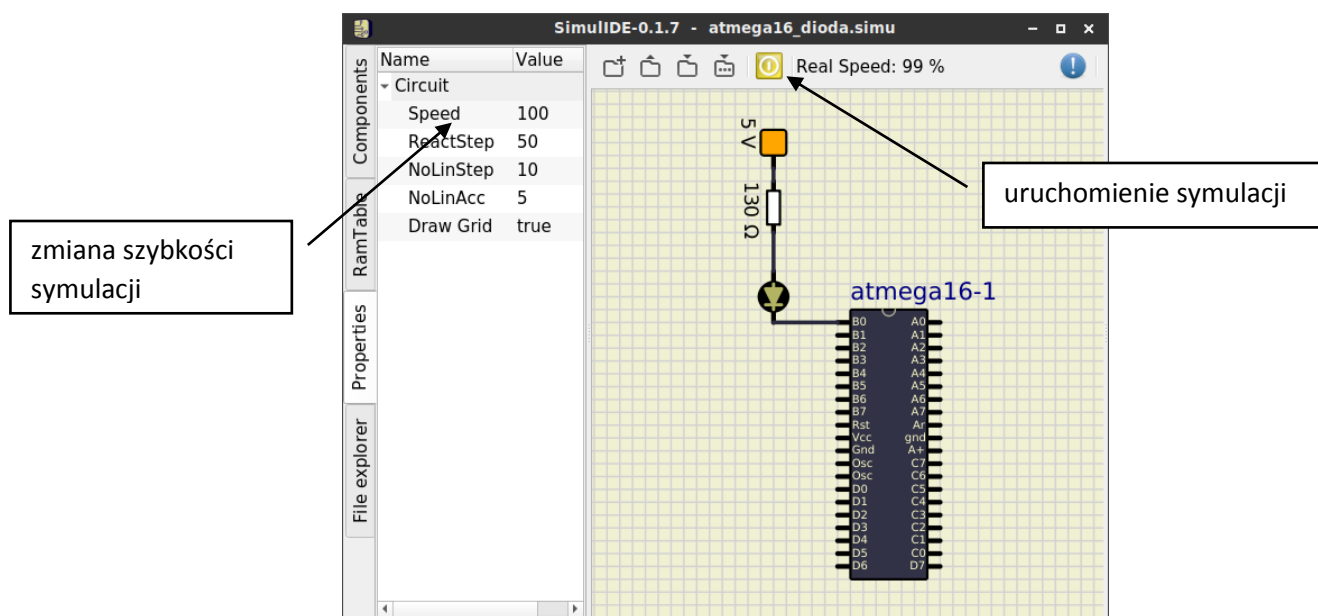


Wartość 20mA to typowa maksymalna wartość prądu obciążenia dla pojedynczej linii I/O mikrokontrolera ATmega16. Dobierając diodę LED w innym kolorze należy skorygować obliczenia, w zależności od spadku napięcia na diodzie  $V_F$  dla wybranej wartości prądu  $I_F$ .

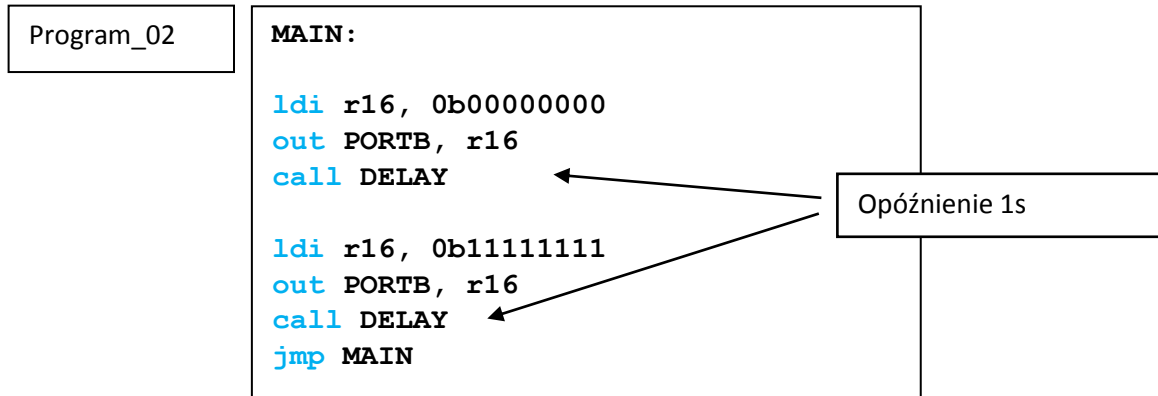
"Wgraj" do procesora plik \*.hex (plik wynikowy wygenerowany podczas pracy w środowisku AVR Studio). Na schemacie zaznacz element procesora, (**Prawy-Przycisk-Myszy > Loadfirmware**). Ustaw odpowiednią częstotliwość pracy procesora (**Prawy-Przycisk-Myszy > Properties > MHz**).

Uruchom symulację w czasie rzeczywistym, dlaczego nie widać mrugania diody LED?

Przypomnij sobie jaką częstotliwość ma generowana fala prostokątna. Bezwładność naszego oka powoduje, że nie jesteśmy w stanie zaobserwować zmian. (W rzeczywistym laboratorium można dołączyć oscyloskop i zaobserwować sygnał). W symulacji można w parametrach programu SimulIDE zmienić domyślną wartość Speed: z 1000000 na 100 (symulacja będzie wykonywana w spowolnieniu).



Uzupełnij program o podprogram DELAY opóźniający działanie o 1s. Wywołaj podprogram po każdej zmianie stanu portu na wyjściu.



Zaobserwuj działanie programu w czasie rzeczywistym (wróć do poprzednich, domyślnych ustawień parametru Speed) w programie SimulIDE. Efekt mrugania diody powinien być widoczny co 1s.

## 6. Zadania do samodzielnej realizacji

### 1. Alternatywne funkcje linii portów

- Na podstawie noty katalogowej ATmega16 opisz w skrócie alternatywne funkcje linii portów A, B, C, D (Tabele 22,25,28,31).

### 2. Programy

- **Zadanie 1:** Napisz program "Biegająca dioda LED". Program ma cyklicznie (co 1s) zmieniać stan portu B, tak aby uzyskać efekt poruszającej się w lewo i w prawo zapalanej diody LED. Zasymuluj działanie w programie SimulIDE podłączając prawidłowo do portu B 8 diod LED w wybranym kolorze. Zmniejsz opóźnienie wprowadzane przez podprogram DELAY, aby przyspieszyć efekt. Podpowiedź: zastosuj instrukcje przesuwania w lewo i w prawo.
- **Zadanie 2:** Napisz program "Magiczne oko" w którym uzyskamy cyklicznie efekt zapalania od środka kolejno 0,2,4 6,8,6,4,2,0.. itd. diod LED podłączonych do portu B. Zasymuluj działanie w programie SimulIDE.
- **Zadanie 3:** Zadanie podane przez prowadzącego.

### 3. Zadania dodatkowe (dla chętnych)

- **Zadanie A:** Napisz program sterujący uproszczoną sygnalizacją świetlną na skrzyżowaniu dróg. Zakładamy, że będą 4 sygnalizatory (po dwa dla każdego kierunku), każdy zawierający światło czerwone i zielone (diody LED) - zatem do sterowania wystarczy 8 linii I/O (jeden port). Zasymuluj działanie w programie SimulIDE. Na potrzeby symulacji okres zmiany świateł ustal co, np. 3 sek. Czy można zmniejszyć liczbę linii I/O stosując układy negatorów (bramki negacji)?
- **Zadanie B:** Zmodyfikuj program Zadanie 1 tak, aby uzyskać efekt pojedynczej "biegnącej diody" na 32 liniowo ustawionych diodach LED podłączonych do portów A, B, C, D procesora. Zasymuluj działanie w programie SimulIDE. Zamiast pojedynczych diod LED i rezystorów możesz na schemacie wstawić symbole zawierające 8 elementów, połączenia można wykonać za pomocą magistral.
- **Zadanie C:** Zadanie podane przez prowadzącego.

## 7. Sprawozdanie

- W sprawozdaniu należy umieścić algorytmy oraz kody programów z odpowiednim wyjaśnieniem działania zastosowanych dyrektyw i instrukcji.
- Na podstawie noty katalogowej ATmega16 sprawdź, jaka jest dopuszczalna wartość prądu pojedynczej linii I/O (Absolute Maximum Ratings, strona 291).