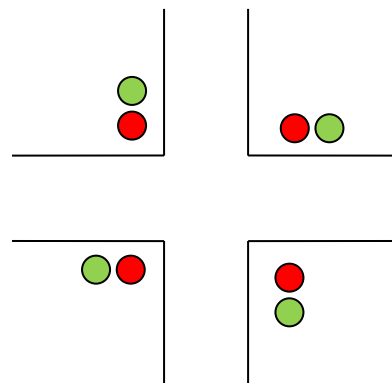


## 1. Zagadnienia

- Zasada działania prostego sterownika świateł
- Algorytm sterowania
- Zastosowanie buforów i inwerterów
- Symulacje w programie SimulIDE

## 2. Zasady sterowanie światłami na skrzyżowaniu

W pierwszej wersji rozważymy sterowanie światłami: czerwonym i zielonym na równorzędnym skrzyżowaniu. Mamy doysterowania 8 świateł - można wykorzystać jeden port I/O (8 linii) mikrokontrolera do niezależnegoysterowania każdego światła. Takie rozwiązanie jest jednak mało efektywne, niesie również spore zagrożenie w przypadku wystąpienia błędów w programie (np. jednoczesne zapalenie wszystkich zielonych świateł). Sposób połączeń powinien minimalizować ryzyko wystąpienia nieodpowiedniej (niebezpiecznej) kombinacji świateł. Dlatego w pojedynczym sygnalizatorze możemy zastosować układ negatora, dzięki któremu zapalimy albo światło zielone, albo czerwone. W podobny sposób - stosując negatory - możemy sterować światłami - gdy jeden kierunek ruchu ma światło zielone, drugi kierunek musi mieć światło czerwone. W ten sposób układ sterowania upraszcza się do wykorzystania 1 linii I/O.



## 3. Program sterownika świateł

Program sterowania światłami dla mikrokontrolera ATmega16 jest stosunkowo prosty. Wysyłamy odpowiednią sekwencję sterującą na linii I/O. W przykładzie Program\_01 sterujemy niezależnie każdym światłem. Spróbuj zasymulować błąd w programie polegający na zapaleniu jednocześnie świateł zielonych i czerwonych.

Na kolejnym schemacie sterowanie sprzętowo zredukowano do jednej linii I/O, co eliminuje powstanie niebezpiecznych błędów. Do celów testowych można zastosować przełącznik, a następnie zastąpić go mikrokontrolerem z odpowiednim oprogramowaniem.

Program\_01

```
// -----
// sterowanie światłami
// -----

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

.cseg
.org 0x0000
rjmp START
;wektory przerwań

.org 0x0030
START:
;ustawienie stosu
ldi r16, low(RAMEND)
out SPL, r16
ldi r16, high(RAMEND)
out SPH, r16

; ustawienie linii I/O
ldi r16, 0b11111111
out DDRB, r16

MAIN:

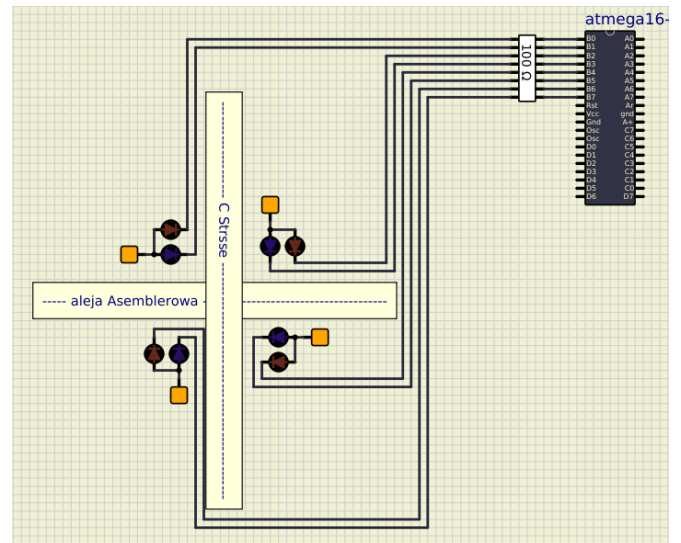
ldi r16, 0b10011001
out PORTB, r16
call DELAY

ldi r16, 0b01100110
out PORTB, r16
call DELAY

jmp MAIN

;dopisz kod podprogramu DELAY
```

wywołanie podprogramu  
opóźnienia, np. 3 sekundy



```
MAIN:
ldi r16, 0b10011001
out PORTB, r16
call DELAY

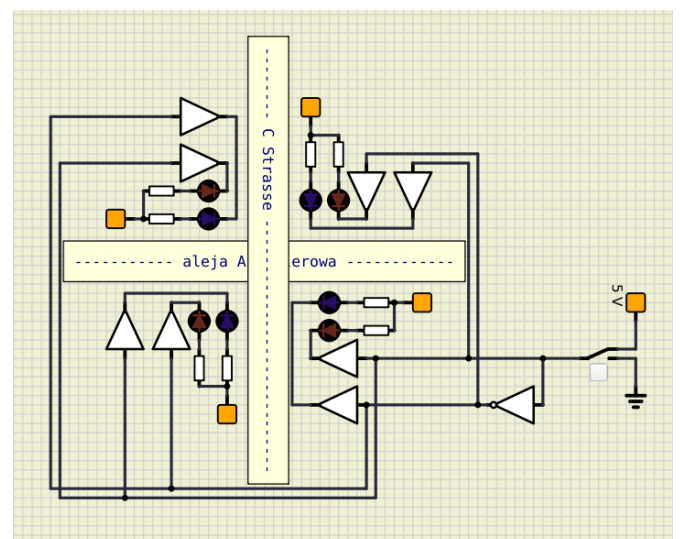
ldi r16, 0b01100000
out PORTB, r16
call DELAY

jmp MAIN
```

symulacja błędu  
programisty

```
MAIN:
cbi PORTB, 0
call DELAY

sbi PORTB, 0
call DELAY
jmp MAIN
```



Na schemacie zastosowano bufory sterujące pracą każdej diody (ze względu na obciążenie prądowe) oraz inwerter. Zastęp przełącznik procesorem ATmega16.

## 4. Program sterownika świateł - wersja rozszerzona

W kolejnej wersji światła na skrzyżowaniu mają dodatkowo niezależnie sterowane światło żółte. Do sterowania wykorzystane zostaną trzy linie I/O.

Program\_02

MAIN:

```
call RED_LIGHT
call YELLOW_LIGHT
call GREEN_LIGHT
```

```
jmp MAIN
```

```
; podprogramy
```

```
.org 0x0100
```

RED\_LIGHT:

```
cbi PORTB, 0
call DELAY_3s
sbi PORTB, 0
ret
```

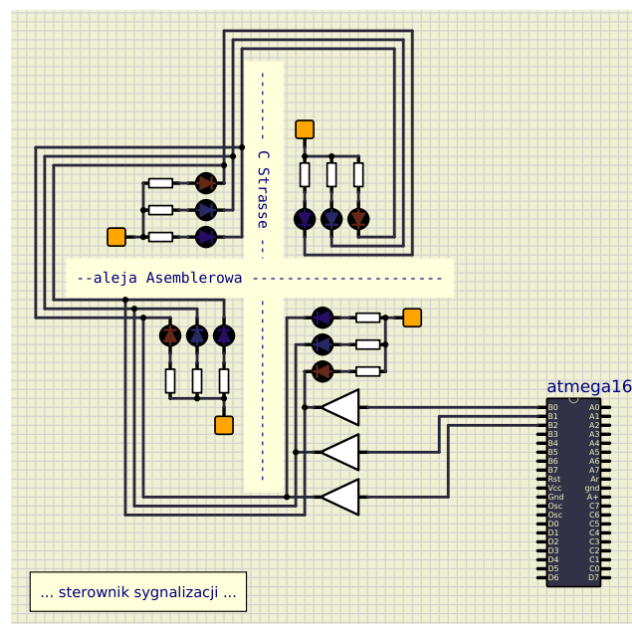
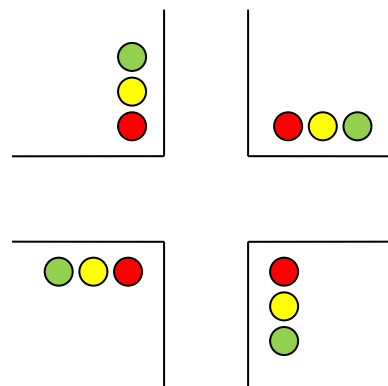
YELLOW\_LIGHT:

```
cbi PORTB, 1
call DELAY_3s
sbi PORTB, 1
ret
```

GREEN\_LIGHT:

```
cbi PORTB, 2
call DELAY_3s
sbi PORTB, 2
ret
```

```
;dopisz kod podprogramu DELAY_3s
```



Takie podłączenie umożliwia niezależne sterowanie sposobem zapalania każdego koloru, jednocześnie eliminuje możliwość jednoczesnego zapalenia światła zielonego dla obu kierunków ruchu.

## 5. Zadania do samodzielnej realizacji

### 1. Programy

- **Zadanie 1:** Zmodyfikuj podprogram dla światła żółtego YELLOW\_LIGHT tak, aby światło żółte mrugało przez 3s (okres mrugania 0.5s). Zasymuluj działanie w programie SimulIDE. Wskazówka: napisz podprogram opóźnienia na 0.5s i kilkakrotnie zapalaj i gaś światło żółte w podprogramie.
- **Zadanie 2:** Uzupełnij schemat sterownika świateł o dodatkowy przycisk AWARIA dołączony do kolejnej linii I/O. Załączenie przycisku powinno załączyć pulsacyjną pracę światła żółtego. Wskazówka: w programie głównym analizuj stan przycisku – wykorzystaj rozkaz warunkowy, lub zastosuj przerwanie zewnętrznym sygnałem.
- **Zadanie 3:** Zadanie podane przez prowadzącego.

### 2. Zadania dodatkowe (dla chętnych)

- **Zadanie A:** Zmodyfikuj schemat układu sterownika tak, aby zapewnić oddzielne sterowanie światłami: czerwonym oraz zielonym dla jednego i drugiego kierunku ruchu, sterowanie światłem żółtym - wspólne, (konieczne 5 linii I/O). Napisz program sterujący tak, aby cykl pracy był następujący:
  - a) światło czerwone świeci przez 8s, (w tym czasie drugi kierunek realizuje cykl dla światła zielonego - (c)),
  - b) światło żółte świeci przez 3s
  - c) światło zielone świeci przez 5s, następnie przez 3s mruga (w tym czasie drugi kierunek realizuje cykl światła czerwonego (a)),
  - d) światło żółte świeci przez 3s, itd...Zasymuluj działanie w programie SimulIDE.
- **Zadanie B:** Zadanie podane przez prowadzącego.

## 6. Sprawozdanie

- W sprawozdaniu należy umieścić algorytmy oraz kody programów z odpowiednim wyjaśnieniem działania zastosowanych dyrektyw i instrukcji.