

**Uniwersytet w Siedlcach  
Wydział Nauk Ścisłych  
Kierunek Informatyka**

**Przedmiot: Architektura Systemów Komputerowych**

***Dokumentacja projektu***

**ŚWIATŁA DROGOWE**

**Opracował: Jakub Michalik, grupa 3,  
I rok informatyki,  
studia stacjonarne inżynierskie**

**Prowadzący:  
Mgr. Inż Maciej Nazarczuk**

**Siedlce, rok akademicki 2024/2025, semestr zimowy**

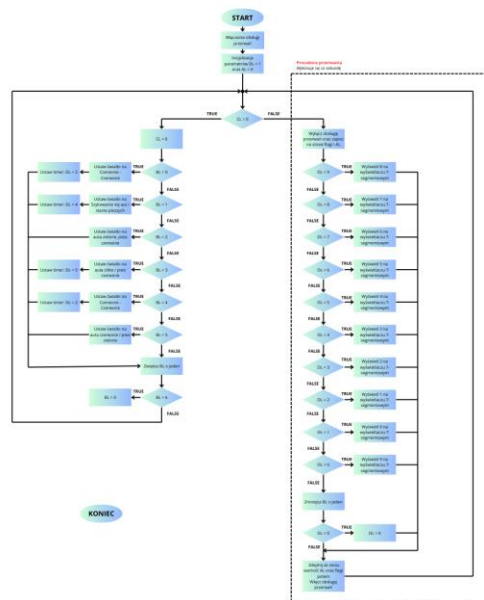
---

# Dokumentacja projektu świateł w SMS32 – Jakub Michalik

## Wprowadzenie

Niniejszy projekt dotyczy implementacji sterowania sygnalizacją świetlną w języku assemblera SMS32. Program obsługuje zmieniające się sygnały świetlne na podstawie wartości w rejestrach, wykorzystując przerwania i procedury do zmiany stanów świateł na pasach (piesi i samochody).

## Schemat blokowy



Schemat blokowy znajduje się w pliku [ŚwiatłaProjekt\\_SchematBlokowy.pdf](#)

## Opis działania programu i przyjęte założenia

Program steruje sygnalizacją świetlną za pomocą rejestrów i instrukcji warunkowych. Główne elementy programu to:

1. Wektory przerwań: Wektor przerwania zegara, który pozwala na manipulację stanami świateł w odpowiednich momentach.

2. Procedury sterujące: Użycie instrukcji CLI (zatrzymanie przerw) i STI (włączenie przerw) do kontrolowania dostępu do przerw w czasie wykonywania programu.

3. Zmienność stanów świateł: Program kontroluje zmiany stanów świateł w zależności od wartości w rejestrach, zwykle w odpowiedzi na zmiany w timerze.

4. Komunikacja z urządzeniami: Przy pomocy instrukcji OUT, dane są wysyłane do portów, aby zmienić stan świateł.

5. Wszystko się powtarza w pętli.

Timer działa tylko w oparciu o liczby jednocyfrowe.

Ograniczenie czasowe zmiany: odliczanie od 9 do 0 (w sekundach)

Sekwencje świateł są zmieniane automatycznie.

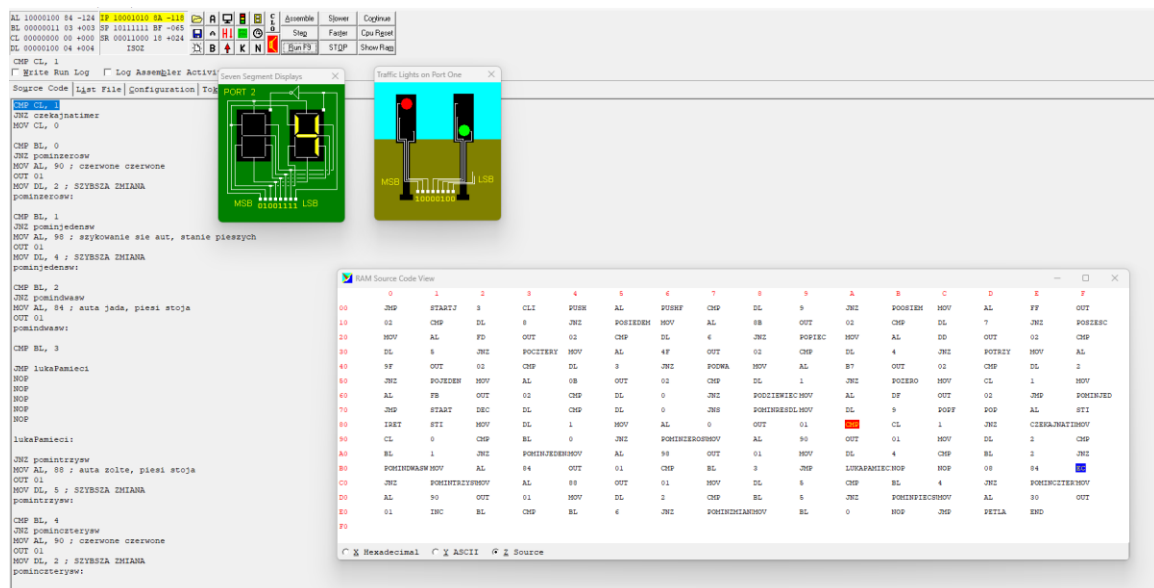
Program się nigdy nie kończy ze względu na nieskończoną pętlę.

Warto zwrócić uwagę na różne odległości czasowe między poszczególnymi stanami światła.

## Prezentacja interfejsu w programie

Na załączonym obrazku widać trzy główne okna programu odpowiadające kolejno (od lewej) za timer, światła pieszych (lewe) i światła aut na ulicy (prawe).

Liczba w timerze zmienia się co sekundę. Światła zmieniają się co parę sekund.



1. Wektory przerwań: Wektor przerywania zegara, który pozwala na manipulację stanami świateł w odpowiednich momentach.

Światła mają kolejne sekwencje powtarzane w pierścieniu Z6.

Przykład: [Wyłączone] → 0 → 1 → 2 → 3 → 4 → 5 → 0 → 1 → 2 → 3 → ...

(Na start programu światła są wyłączone)

0. Piesi: Czerwone / Auta: Czerwone (przez 2 sekund)

1. Piesi: Czerwone / Auta: Żółte i Czerwone (przez 4 sekund)

2. Piesi: Czerwone / Auta: Zielone (przez 9 sekund)

3. Piesi: Czerwone / Auta: Żółte (przez 5 sekund)

4. Piesi: Czerwone / Auta: Czerwone (przez 2 sekund)

5. Piesi: Zielone / Auta: Czerwone (przez 9 sekund)

## Kod źródłowy

Poniżej znajduje się pełny kod programu implementującego sterowanie sygnalizacją świetlną.

JMP startj ; Skok za tablice wektorow przerwan

DB 3 ; Wektor przerywania zegara

ORG 3

CLI ; WYLACZENIE PRZERWAN

PUSH AL

PUSHF

CMP DL, 9

JNZ poosiem

MOV AL, FF

OUT 02 ;wyswietleni 8

poosiem:

CMP DL, 8

JNZ posiedem

MOV AL, 8B

OUT 02 ;wyswietleni 7  
posiedem:

CMP DL, 7  
JNZ poszesc  
MOV AL, FD  
OUT 02 ;wyswietleni 6  
poszesc:

CMP DL, 6  
JNZ popiec  
MOV AL, DD  
OUT 02 ;wyswietleni 5  
popiec:

CMP DL, 5  
JNZ pocztery  
MOV AL, 4F  
OUT 02 ;wyświetleni 4  
pocztery:

CMP DL, 4  
JNZ potrzy  
MOV AL, 9F  
OUT 02 ;wyswietleni 3  
potrzy:

CMP DL, 3  
JNZ podwa  
MOV AL, B7  
OUT 02 ;wyswietleni 2  
podwa:

CMP DL, 2  
JNZ pojeden  
MOV AL, 0B  
OUT 02 ;wyswietleni 1  
pojeden:

CMP DL, 1  
JNZ pozero  
MOV CL, 1

```
MOV AL, FB
OUT 02      ;wyswietleni 0
pozero:
```

```
CMP DL, 0
JNZ podziewiec
MOV AL, DF
OUT 02      ;wyswietleni 9
podziewiec:
```

```
JMP pominjed
startj:
JMP start
pominjed:
```

```
DEC DL
CMP DL, 0
JNS pominresdl
MOV DL, 9
pominresdl:
```

```
POPF
POP AL
```

```
STI ; WLACZENIE PRZERWAN
IRET
```

```
start:
STI
MOV DL, 1 ; resetuje timer
```

```
MOV AL, 0
OUT 01
```

```
PETLA:
```

```
CMP CL, 1
JNZ czekajnatimer
MOV CL, 0
```

```
CMP BL, 0
```

JNZ pominzerosw  
MOV AL, 90 ; czerwone czerwone  
OUT 01  
MOV DL, 2 ; SZYBSZA ZMIANA  
pominzerosw:

CMP BL, 1  
JNZ pominjedensw  
MOV AL, 98 ; szykowanie sie aut, stanie pieszych  
OUT 01  
MOV DL, 4 ; SZYBSZA ZMIANA  
pominjedensw:

CMP BL, 2  
JNZ pomindwasw  
MOV AL, 84 ; auta jada, piesi stoja  
OUT 01  
pomindwasw:

CMP BL, 3

JMP lukaPamieci  
NOP  
NOP  
NOP  
NOP  
NOP

lukaPamieci:

JNZ pomintrzysw  
MOV AL, 88 ; auta zolte, piesi stoja  
OUT 01  
MOV DL, 5 ; SZYBSZA ZMIANA  
pomintrzysw:

CMP BL, 4  
JNZ pominczterysw  
MOV AL, 90 ; czerwone czerwone  
OUT 01  
MOV DL, 2 ; SZYBSZA ZMIANA  
pominczterysw:

CMP BL, 5  
JNZ pominpiecsw  
MOV AL, 30 ; auta staja, piesi ida  
OUT 01  
pominpiecsw:

INC BL  
CMP BL,6  
JNZ pominzmianebl  
MOV BL,0  
pominzmianebl:

czekajnatimer:  
NOP  
JMP PETLA

END

## Spis Treści \*

### 1. Strona tytułowa

### 2. Wprowadzenie

- Cel projektu
- Zakres realizacji

### 3. Schemat blokowy

- Lokalizacja pliku schematu blokowego

### 4. Opis działania programu i przyjęte założenia

- Wektory przerwań
- Procedury sterujące
- Zmienność stanów świateł
- Komunikacja z urządzeniami
- Timer i odliczanie czasu
- Sekwencje świateł

### 5. Prezentacja interfejsu w programie

- Opis głównych okien programu
- Zmiany stanu świateł i timera



## **6. Przykład sekwencji świateł**

- Kolejne stany świateł w pierścieniu Z6
- Szczegółowe czasy trwania poszczególnych stanów

## **7. Kod źródłowy**

- Pełna implementacja programu w języku SMS32

\*Spis treści został wygenerowany za pomocą ChatGPT z modelem v2