

Projektowanie oprogramowania wbudowanego

Przykład – sterowanie oświetleniem sali wykładowej

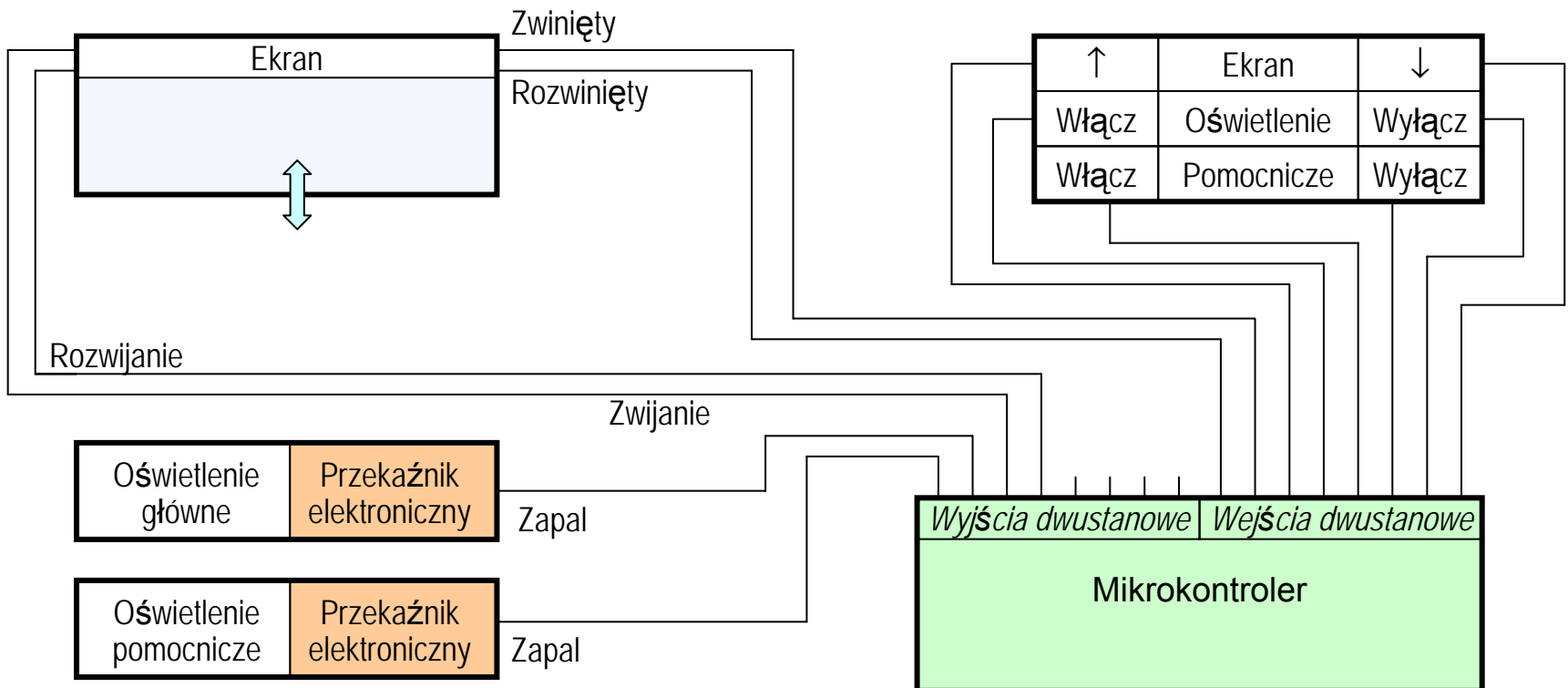
Wyposażenie sali składa się z elektrycznie podnoszonego ekranu, świateł głównych i pomocniczych oraz pulpitu ręcznego sterowania.

Wymagane funkcje systemu obejmują:

1. Automatyczne rozwijanie i zwijanie ekranu, sterowane przez użytkownika za pomocą 2 przycisków pulpitu.
2. Płynne zapalanie i gaszenie świateł głównych w ciągu 10s, z możliwością zatrzymania wybranego poziomu świecenia, sterowane przez użytkownika za pomocą 2 przycisków pulpitu.
3. Zapalanie i gaszenie świateł pomocniczych, sterowane przez użytkownika za pomocą 2 przycisków pulpitu.

Projektowanie systemu

Schemat instalacji



Projekt sprzętu

1. Wybór mikrokontrolera, np. 80C51 / 12 MHz
2. Projekt sprzęgu procesowego
3. Przypisanie sygnałów wejściowych i wyjściowych

Wejścia (port P1)

E0 – start rozwijania ekranu
E1 – wyłączenie świateł głównych
E2 – wyłączenie świateł pomocniczych
E3 – włączenie świateł pomocniczych
E4 – włączenie świateł głównych
E5 – start zwijania ekranu
E6 – ekran zwinięty
E7 – ekran rozwinięty

Wyjścia (port P2)

Y0
Y1
Y2
Y3
Y4 – rozwijanie ekranu
Y5 – zwijanie ekranu
Y6 – zapalenie świateł głównych
Y7 – zapalenie świateł pomocniczych

Spostrzeżenia

1. System musi działać współbieżnie
2. Reakcje na zdarzenia zależą od stanu systemu
3. Istnieją zabronione kombinacje wejść

Ekran rozwinięty i jednocześnie ekran zwinięty

4. Kombinacja wejść E7 & E6 oznacza awarię – sygnalizacja:
Y0 – zapalenie czerwonej diody na pulpicie

Analiza wymagań

Lista zdarzeń

- 1.1 Naciśnięty przycisk rozwijania ekranu
- 1.2 Naciśnięty przycisk zwijania ekranu
- 1.3 Ekran rozwinięty
- 1.4 Ekran zwinięty

- 2.1 Naciśnięty przycisk włączenia świateł
- 2.2 Naciśnięty przycisk wyłączenie świateł
- 2.3 Pełne zapalenie świateł
- 2.4 Pełne zgaszenie świateł

- 3.1 Włączenie świateł pomocniczych
- 3.2 Wyłączenie świateł pomocniczych

1. Automatyczne rozwijanie i zwijanie ekranu, sterowane przez użytkownika za pomocą 2 przycisków pulpitu.

2. Płynne zapalenie i gaszenie świateł głównych w ciągu 10s, z możliwością zatrzymania wybranego poziomu świecenia, sterowane przez użytkownika za pomocą 2 przycisków pulpitu.

3. Zapalenie i gaszenie świateł pomocniczych, sterowane przez użytkownika za pomocą 2 przycisków pulpitu.

Określenie reakcji na zdarzenia

Nr	Zdarzenie	Reakcja
1.1	Przycisk rozwijania ekranu	Jeżeli ekran jest nieruchomy, to włączyć silnik ekranu w dół. Jeżeli ekran się zwija , to wyłączyć silnik Jeżeli ekran jest rozwinięty , to brak reakcji. Jeżeli naciśnięty przycisk zwijania , to brak reakcji.
1.2	Przycisk zwijania ekranu	
1.3	Ekran rozwinięty	
1.4	Ekran zwinięty	
2.1	Przycisk włączenia świateł	
2.2	Przycisk wyłączenia świateł	
2.3	Pełne zapalenie świateł	
2.4	Pełne wygaszenie świateł	
3.1	Włączenie świateł pomocniczych	
3.2	Wyłączenie świateł pomocniczych	

Określenie reakcji na zdarzenia

Nr	Zdarzenie	Reakcja
1.1	Przycisk rozwijania ekranu	Jeżeli ekran jest nieruchomy, to włączyć silnik ekranu w dół. Jeżeli ekran się zwija , to wyłączyć silnik. Jeżeli ekran jest rozwinęty , to brak reakcji. Jeżeli naciśnięty przycisk zwijania , to brak reakcji.
1.2	Przycisk zwijania ekranu	----- <i>analogicznie</i> -----
1.3	Ekran rozwinęty	Wyłączyć silnik ekranu. Jeżeli jednocześnie ekran zwinęty , to sygnalizować awarię
1.4	Ekran zwinęty	----- <i>analogicznie</i> -----
2.1	Przycisk włączenia świateł	Uruchomić generację narastającej fali PWM. Jeżeli światła zapalone , to brak reakcji. Jeżeli światła opadają , to zatrzymać opadanie. Jeżeli naciśnięty przycisk wyłączenia , to brak reakcji.
2.2	Przycisk wyłączenia świateł	----- <i>analogicznie</i> -----
2.3	Pełne zapalenie świateł	Zatrzymać generację fali, pełne włączenie świateł
2.4	Pełne wygaszenie świateł	Zatrzymać generację fali, pełne wygaszenie świateł
3.1	Włączenie świateł pomocniczych	Zapalenie świateł Jeżeli naciśnięty przycisk wyłączenia , to brak reakcji
3.2	Wyłączenie świateł pomocniczych	----- <i>analogicznie</i> -----

Określenie ograniczeń czasowych

1. Okres taktowania fali PWM 2ms.
2. Czas reakcji na zadziałanie czujnika krańcowego $\leq 10\text{ms.}$
3. Czas reakcji na naciśnięcie przycisku $\leq 100\text{ms.}$

Ad 1) Nowe zdarzenie – *Takt PWM*

Ad 2) Reakcja na zadziałanie czujników jest reakcją na zdarzenia:
Ekran rozwinięty, Ekran zwinięty

Ad 3) Reakcja na naciśnięcie przycisków dotyczy zdarzeń:
*Przycisk rozwijania ekranu, Przycisk zwijania ekranu,
Przycisk włączenia świateł, Przycisk wyłączenia świateł,
Włączenie pomocniczych, Wyłączenie pomocniczych*

Wymagania dla oprogramowania

1. Realizacja z użyciem mikrokontrolera 80C51 / 12 MHz
2. Wymagania funkcjonalne podane w tabeli reakcji zdarzeń
3. Wymagania czasowe podane w tabeli
4. Wymagania bezpieczeństwa:
 - 4.1. Nie wolno jednocześnie wysterować zwijania i rozwijania ekranu (konieczna przerwa min. 1ms)
 - 4.2. Brak sygnału z wyłącznika krańcowego po 12s pracy silnika zatrzymuje silnik i uruchamia sygnał awarii
 - 4.3. Zatrzymanie aplikacji (przekroczenie okresu PWM o 0,5ms) powoduje wyłączenie silnika ekranu i restart systemu

Określenie zadań i ich ograniczeń czasowych

Nr	Zdarzenie	Reakcja
1.1	Przycisk rozwijania ekranu	Jeżeli ekran jest nieruchomy, to włączyć silnik ekranu w dół. Jeżeli ekran się zwija, to wyłączyć silnik. Jeżeli ekran jest rozwinięty, to brak reakcji. Jeżeli naciśnięty przycisk zwijania, to brak reakcji.
1.2	Przycisk zwijania ekranu	----- <i>analogicznie</i> -----
1.3	Ekran rozwinięty	Wyłączyć silnik ekranu. Jeżeli jednocześnie ekran zwinięty, to sygnalizować awarię
1.4	Ekran zwinięty	----- <i>analogicznie</i> -----
2.1	Przycisk włączenia świateł	Uruchomić generację narastającej fali PWM. Jeżeli światła zapalone, to brak reakcji. Jeżeli światła opadają, to zatrzymać opadanie. Jeżeli naciśnięty przycisk wyłączenia , to brak reakcji.
2.2	Przycisk wyłączenia świateł	----- <i>analogicznie</i> -----
2.3	Pełne zapalenie świateł	Zatrzymać generację fali, pełne włączenie świateł
2.4	Pełne wygaszenie świateł	Zatrzymać generację fali, pełne wygaszenie świateł
2.5	Takt fali PWM	Generacja sygnału PWM sterującego mocą świateł głównych
3.1	Włączenie świateł pomocniczych	Zapalenie świateł Jeżeli naciśnięty przycisk wyłączenia , to brak reakcji
3.2	Wyłączenie świateł pomocniczych	----- <i>analogicznie</i> -----

Określenie zadań i ich ograniczeń czasowych

Nr	Zdarzenie	Zadanie	Okres
1	Przycisk rozwijania ekranu Przycisk zwijania ekranu	Start ekranu Obsługa przycisków uruchamiających rozwijanie i zwijanie ekranu. Uruchomienie odliczania czasu ruchu silnika.	≤ 100 ms
2	Ekran rozwinięty Ekran zwinięty	Stop ekranu Obsługa czujników krańcowych zatrzymujących rozwijanie i zwijanie ekranu. Obsługa awarii.	≤ 10 ms
3	Przycisk włączenia świateł Przycisk wyłączenia świateł	Sterowanie światłami głównymi Obsługa przycisków zapalających i gaszących światła główne.	≤ 100 ms
4	Pełne zapalenie świateł Pełne wygaszenie świateł Takt fali PWM	Generacja fali PWM Sterowanie narastaniem i opadaniem świateł.	2 ms
5	Włączenie świateł pomocniczych Wyłączenie świateł pomocniczych	Sterowanie światłami pomocniczymi Obsługa przycisków zapalających i gaszących światła pomocnicze.	≤ 100 ms

Określenie struktury aplikacji

Nr	Zadanie	0	1	Okres [ms]	Czas wykonania [μ s]
1	Start ekranu	×		≤ 100	500
2	Stop ekranu	×		≤ 10	500
3	Sterowanie światłami gł		×	≤ 100	500
4	Generacja fali PWM	×	×	2	500
5	Sterowanie światłami po		×	≤ 100	500
		1,5	1,5		

Wniosek: organizacja sekwencyjna, stałe rozplanowanie zadań

Zadanie 1: Start ekranu

Wyjście (Y)		Wejście (E)		Akcja
Z	R	Z	R	
0	0	0	0	Nie rób nic
0	0	0	1	Ustaw bit rozwijania ekranu / Pomiar 12s
0	0	1	0	Ustaw bit zwijania ekranu / Pomiar 12s
0	0	1	1	Nie rób nic
0	1	0	0	Nie rób nic
0	1	0	1	Nie rób nic
0	1	1	0	Zeruj bit rozwijania
0	1	1	1	Nie rób nic
1	0	0	0	Nie rób nic
1	0	0	1	Zeruj bit zwijania
1	0	1	0	Nie rób nic
1	0	1	1	Nie rób nic
1	1	x	x	??

Zadanie 1: Start ekranu

```

unsigned char e,y;
unsigned short t12;

void start_screen()
{
    unsigned char z;
    z = e & 0x21;
    switch ( y & 0x30 ) {
    case 0x00:
        if ( z == 0x01 ) y = y | 0x10;
        if ( z == 0x20 ) y = y | 0x20;
        t12 = 0;
        break
    case 0x10:
        if ( z == 0x20 ) y = y & 0xEF;
        break;
    case 0x20:
        if ( z == 0x01 ) y = y & 0xDF;
        break;
    default:
        y = y & 0xCF;
    }
}

```

Komunikacja z otoczeniem:

e		Z					R
y		Z	R				A
t12							

e		y		Akcja
Z	R	Z	R	
0	1	0	0	Rozwijaj
1	0	0	0	Zwijaj
1	0	0	1	Zeruj rozwijanie
0	1	1	0	Zeruj zwijanie

Zadanie 2: Stop ekranu

```

unsigned char e,y;
unsigned short t12;

void stop_screen()
{
    unsigned char z;
    z = ( e & 0xC0 ) | ( y & 0x30 );
    if ( z == 0x90 ) y = y & 0xEF;
    if ( z == 0x60 ) y = y & 0xDF;
    if ( t12 > 6000 ) y = y | 0x01;
    if ( (e & 0xC0) == 0xC0 ) y = y | 0x01;
    if ( (y & 0x01) == 0x01 ) y = y & 0xCF;
}
    
```

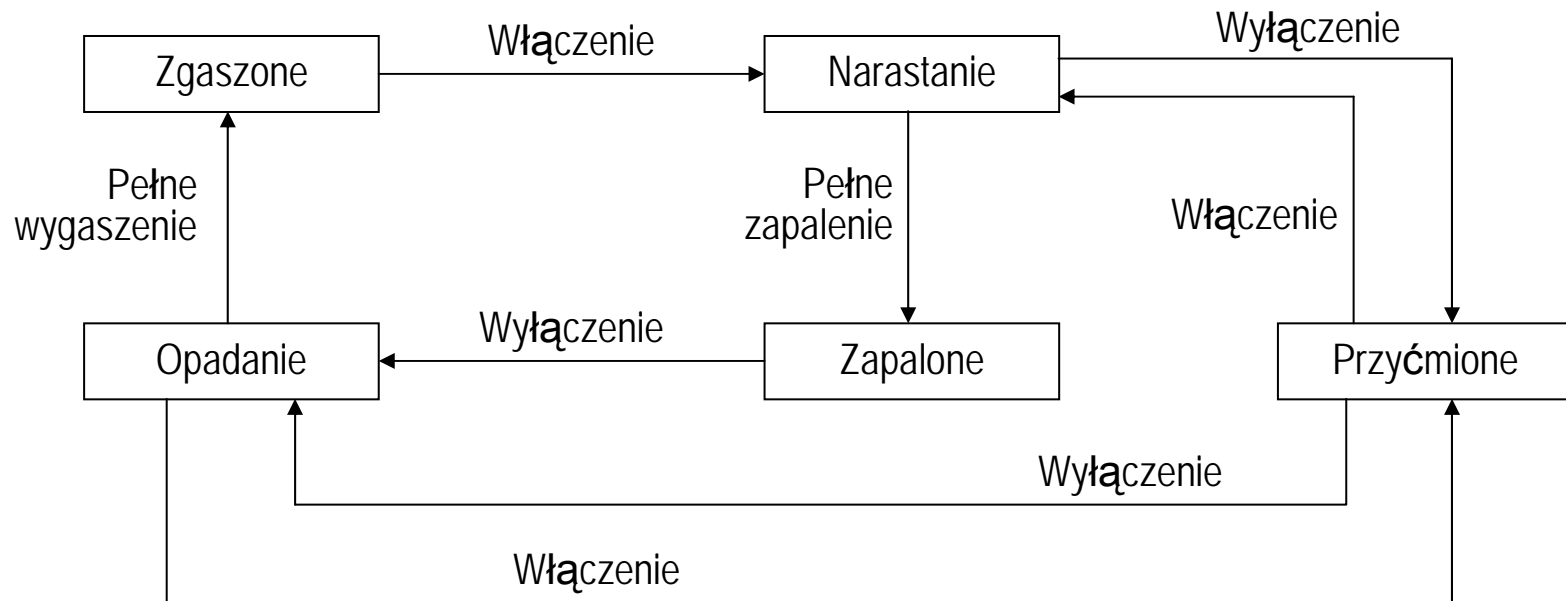
Komunikacja z otoczeniem:

e	r	z					
y			Z	R			A
t12							

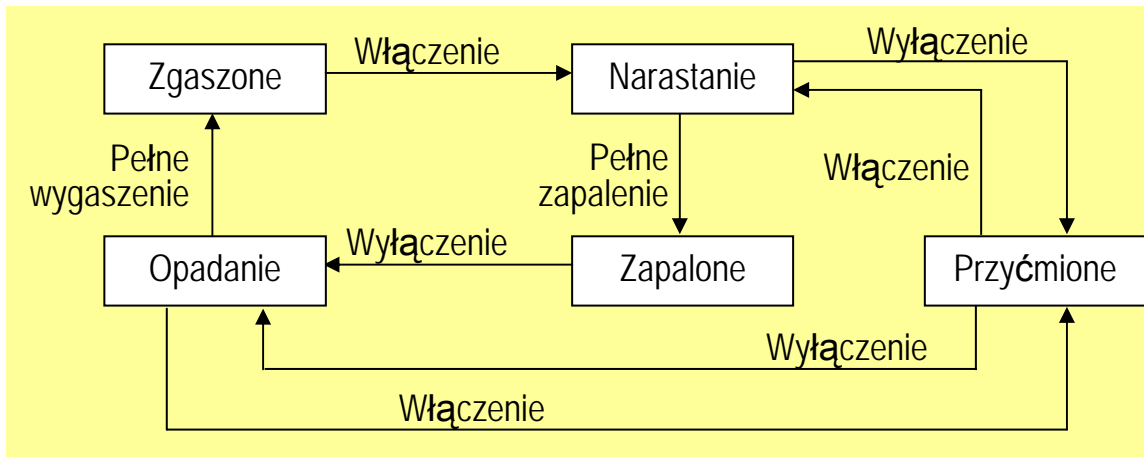
e		y		Akcja
r	z	Z	R	
x	x	0	0	Nie rób nic
1	0	0	1	Zeruj rozwijanie
0	1	1	0	Zeruj zwijanie
1	1	x	x	Awaria / Zeruj

Zadanie 3: Sterowanie światłami głównymi

Tryby pracy świateł (*modes of control*)



Zadanie 3: Sterowanie światłami głównymi



	Włącz	Wyłącz
Zgaszone (0)	Narastanie	-
Narastanie (1)	-	Przyćmione
Zapalone (2)	-	Opadanie
Opadanie (3)	Przyćmione	-
Przyćmione (4)	Narastanie	Opadanie

G	-G	Zdarzenie
0	0	Nie rób nic
0	1	Wyłącz
1	0	Włącz
1	1	Nie rób nic

Zadanie 3: Sterowanie światłami głównymi

```

unsigned char e;
unsigned char s;

void main_lights()
{
    unsigned char z;
    z = ( e & 0x12 );
    switch ( s ) {
    case 0:
        if ( z == 0x10 ) s = 1;
        break;
    case 1:
        if ( z == 0x02 ) s = 4;
        break;
    case 2:
        if ( z == 0x02 ) s = 3;
        break;
    case 3:
        if ( z == 0x10 ) s = 4;
        break;
    case 4:
        if ( z == 0x10 ) s = 1;
        if ( z == 0x02 ) s = 3;
    }
}

```

Komunikacja z otoczeniem:

e				G			-G	
s						S	S	S

	Włącz (G)	Wyłącz (-G)
Zgaszone (0)	Narastanie	-
Narastanie (1)	-	Przyćmione
Zapalone (2)	-	Opadanie
Opadanie (3)	Przyćmione	-
Przyćmione (4)	Narastanie	Opadanie

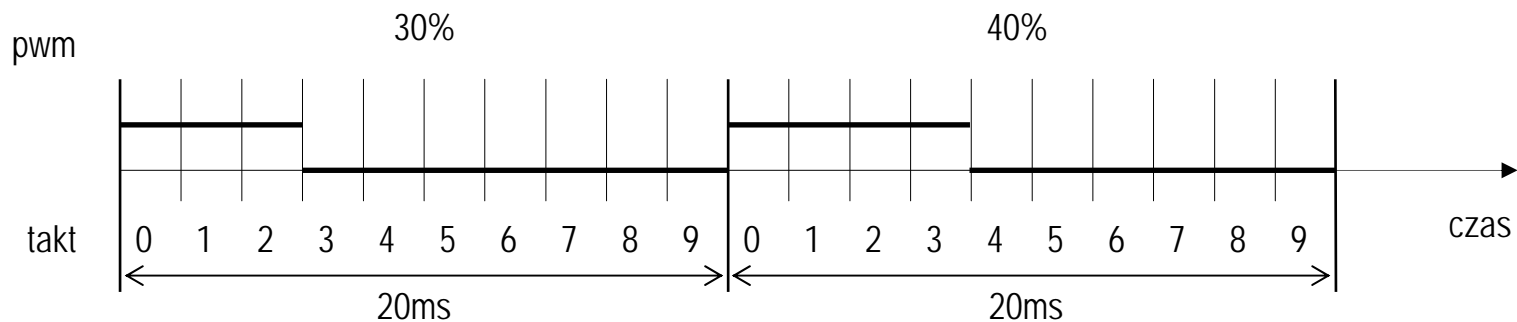
Zadanie 4: Generacja fali PWM

```

takt++;
if ( takt == 10 ) { takt = 0; okres++; }
if ( okres == 50 ) { okres = 0; pwm++; }
if ( pwm == 10 )
    { s = Zapalone; G = 1; }
else
    if ( takt < pwm ) G = 1; else G = 0;
    
```

Komunikacja z otoczeniem:

y		G						
s						S	S	S
pwm								
takt								
okres								



Zadanie 4: Generacja fali PWM

```

unsigned char y;
unsigned char s;
unsigned char pwm, takt, okres;

void pwm()
{
    takt++;
    if ( takt == 10 ) takt = 0;
    if ( takt < pwm ) y = y | 0x40; else y = y & 0xBF;
    if ( s == 1 ) { // narastanie
        if ( takt == 0 ) okres++;
        if ( okres == 50 ) { okres = 0; pwm++; }
        if ( pwm == 10 ) s = 2; // zapalone
    }
    if ( s == 3 ) { // opadanie
        if ( takt == 0 ) okres++;
        if ( okres == 50 ) { okres = 0; pwm--; }
        if ( pwm == 0 ) s = 0; // zgaszone
    }
}

```

Komunikacja z otoczeniem:

y	G						
s					S	S	S
pwm							
takt							
okres							

Program główny

```

unsigned char  e,y,s;
unsigned char  i;
unsigned char  pwm, takt, okres;
unsigned short t12;

void main()
{
    init();
    while ( 1 ) {
        out8(P2,y);
        while ( in8(TH0)<240 );
        out8(TH0,178);
        e = in8(P1);
        if ( i==0) start_screen( );
        if ( i==0) stop_screen( );
        pwm( );
        if ( i==1) main_lights( );
        if ( i==1) aux_lights( );
        t12++;
        i = (i+1)%2;
    }
}

```

Komunikacja z otoczeniem:

e	r	z	Z	G	P	-P	-G	R	
y	P	G	Z	R				A	
s						S	S	S	
t12									
TMOD	G1	C/T	M1	M0	G0	C/T	M1	M0	
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
TH0									
TL0							X	X	X

240 - 178 = 62

62 × 32 = 1984 μs

```

void init()
{
    y = 0;           // wyłączone
    s = 0;           // zgaszone
    i = 0;           // kwant 0
    pwm = 0;
    takt = 0;
    okres = 0;
    out8(TM0D,0);
    out8(TH0,178);
    out8(TCON,0x10);
}

```