

Projektowanie oprogramowania wbudowanego

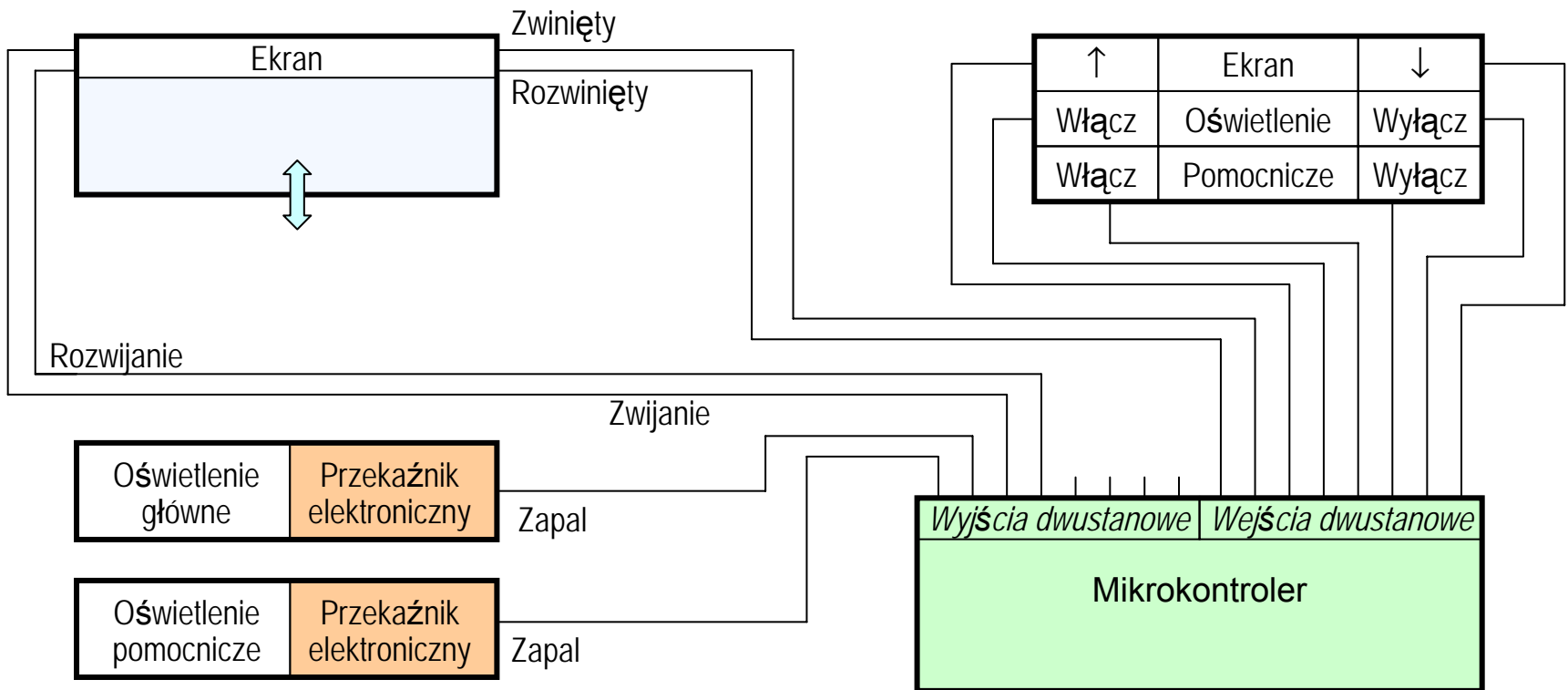
Przykład wyposażenie sali wykładowej składa się z ekranu, świateł głównych i pomocniczych oraz pulpitu ręcznego sterowania.

Zaprojektować system realizujący następujące funkcje :

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



Bezpieczeństwo

- wymagania urzędzeń
 - awarie
-
-

Projekt sprzętu

1. Wybór mikrokontrolera.
2. Przypisanie sygnałów wejściowych i wyjściowych.
3. Projekt sprzęgu procesowego.

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
- 2.5 Upłynął takt fali PWM

- 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 się rozwija lub 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	
2.5	Takt fali PWM	
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 się rozwija lub 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 przygasają, to zatrzymać opadanie. Jeżeli światła zapalają się lub są zapalone, to brak reakcji. Jeżeli naciśnięty przycisk wyłączenia , to brak reakcji.
2.2	Przycisk wyłączenia świateł	----- <i>analogicznie</i> -----
2.3	Pełne zapalenie świateł	Zatrzymać narastanie fali PWM (pełne włączenie świateł)
2.4	Pełne wygaszenie świateł	Zatrzymać opadanie fali PWM (pełne wygaszenie świateł)
2.5	Takt fali PWM	Generacja narastającej stałej lub opadającej fali PWM
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

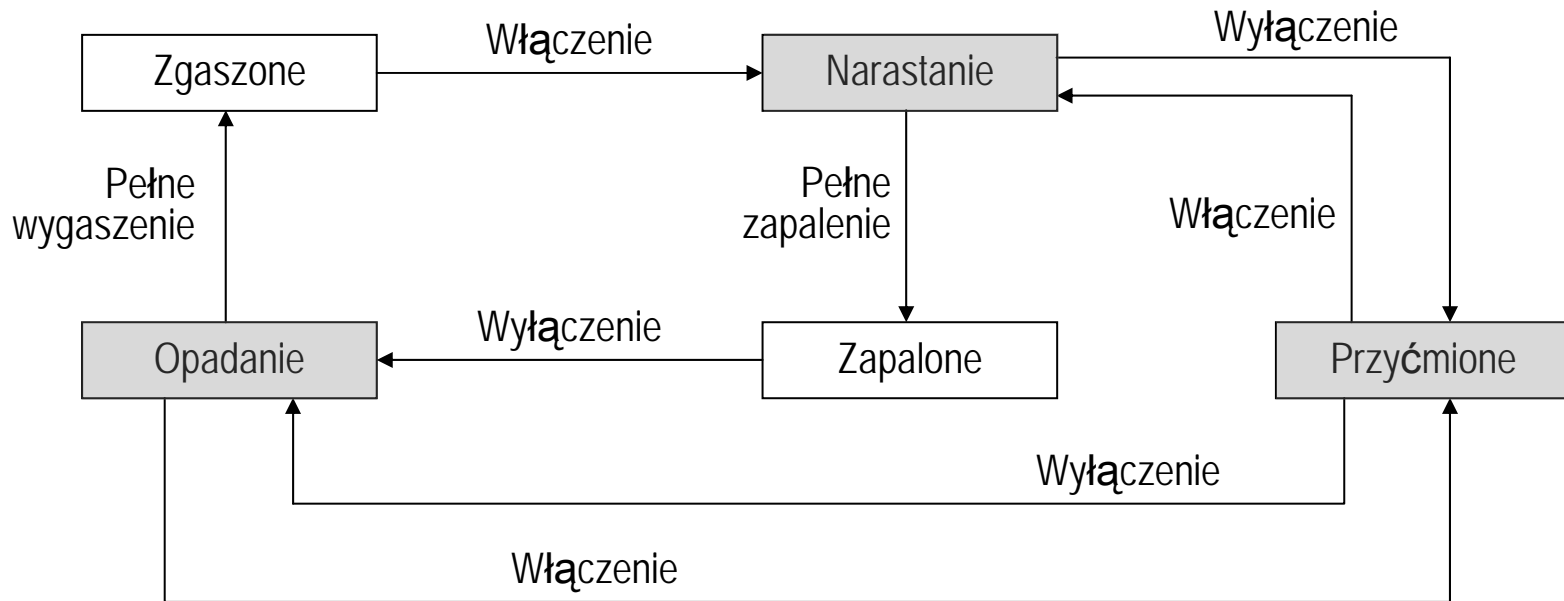
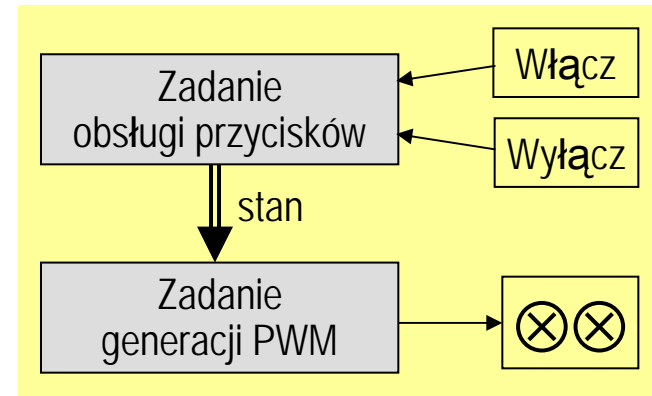
1. Okres taktowania fali PWM 2ms.
2. Czas reakcji na zadziałanie czujnika krańcowego $\leq 10\text{ms.}$
3. Czas reakcji na naciśnięcie przycisku $\leq 100\text{ms.}$

Projektowanie struktury aplikacji

Nr	Zdarzenie	Reakcja	Okres
1.1	Przycisk rozwijania ekranu	Jeżeli ekran jest nieruchomy, to włączyć silnik ekranu w dół. Jeżeli ekran się zwija , to wyłączyć silnik. Jeżeli ekran się rozwija lub jest rozwinięty , to brak reakcji. Jeżeli naciśnięty przycisk zwijania , to brak reakcji.	100ms
1.2	Przycisk zwijania ekranu	----- <i>analogicznie</i> -----	100ms
1.3	Ekran rozwinięty	Wyłączyć silnik ekranu. Jeżeli jednocześnie ekran zwinięty , to sygnalizować awarię	10ms
1.4	Ekran zwinięty	----- <i>analogicznie</i> -----	10ms
2.1	Przycisk włączenia świateł	Uruchomić generację narastającej fali PWM. Jeżeli światła przygasają , to zatrzymać opadanie. Jeżeli światła zapalają się lub są zapalone , to brak reakcji. Jeżeli naciśnięty przycisk wyłączenia , to brak reakcji.	100ms
2.2	Przycisk wyłączenia świateł	----- <i>analogicznie</i> -----	100ms
2.3	Pełne zapalenie świateł	Zatrzymać narastanie fali PWM (pełne włączenie świateł)	
2.4	Pełne wygaszenie świateł	Zatrzymać opadanie fali PWM (pełne wygaszenie świateł)	
2.5	Takt fali PWM	Generacja narastającej stałej lub opadającej fali PWM	2ms
3.1	Włączenie świateł pomoc.	Zapalenie świateł Jeżeli naciśnięty przycisk wyłączenia , to brak reakcji	100ms
3.2	Wyłączenie świateł pomoc.	----- <i>analogicznie</i> -----	100ms

Sterowanie światłami głównymi

Tryby pracy świateł (*modes of control*)



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.	≤ 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

Rozplanowanie wykonania zadań

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

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

Zadanie 1: Start ekranu

```

unsigned char e,y;
unsigned short t12;

void start_screen()
{
    unsigned char z;
    z = (e & 0x20)<<2 | (e & 0x01)<<6 | (y & 0x30);
    if ( z == 0x40 ) y = y | 0x10;
    if ( z == 0x80 ) y = y | 0x20;
    if ( z == 0x90 ) y = y & 0xCF;
    if ( z == 0x60 ) y = y & 0xCF;
    if ( (y & 0x30) == 0x00 ) t12=0;
}

```

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	Zatrzymaj silnik
0	1	1	0	Zatrzymaj silnik

Zadanie 2: Stop ekranu

```

unsigned char e,y;
unsigned short t12;

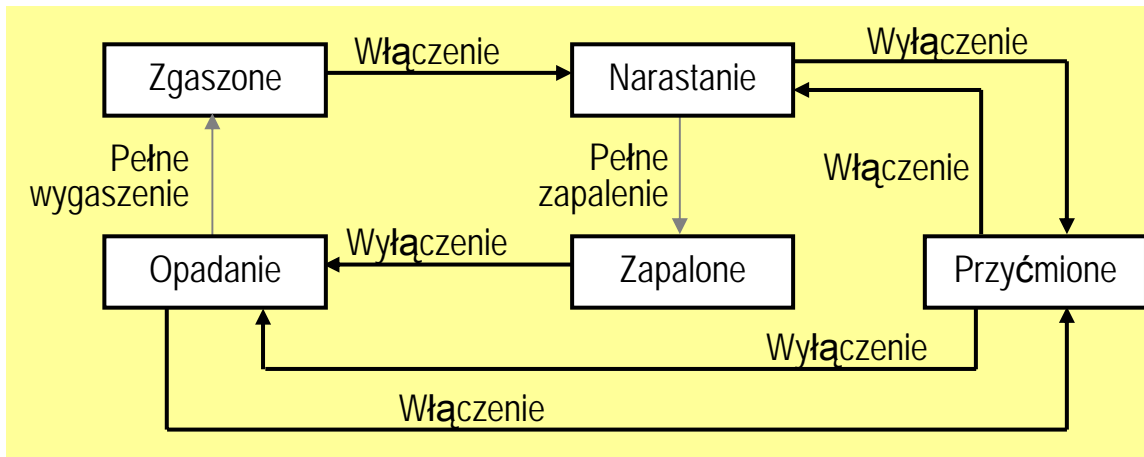
void stop_screen()
{
    unsigned char z;
    z = ( e & 0xC0 ) | ( y & 0x30 );
    if ( z == 0x90 ) y = y & 0xCF;
    if ( z == 0x60 ) y = y & 0xCF;
    if ( t12 >= 6000 ) y = y | 0x01;
    if ( ( e & 0xC0 ) == 0xC0 ) y = y | 0x01;
    if ( ( e & 0x01 ) == 0x01 ) y = y & 0xCF;
}
    
```

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	Zatrzymaj silnik
0	1	1	0	Zatrzymaj silnik
1	1	x	x	Awaria

Zadanie 3: Sterowanie światłami głównymi



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

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

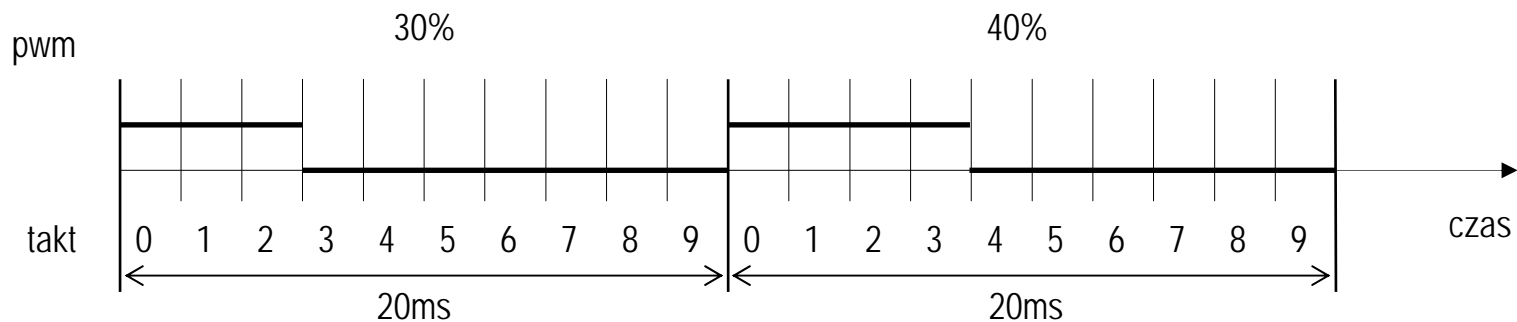
Zadanie 4: Generacja fali PWM

```

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)<62 );
        out8(TH0,0);
        e = in8(P1);
        if ( i==0) start_screen( );
        if ( i==0) stop_screen( );
        if ( i==1) main_lights( );
        if ( i==1) aux_lights( );
        pwm( );
        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

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(TH0,0);
    out8(TCON,0x10);
}

```