

6 cz. 2 Linie I/O, porty wejścia/wyjścia

Imię nazwisko, indeks :
Marcel Garczyk, 147935
 Paweł Hatka, 147952

Data wykonania : 01.12.2022r.

Grupa : T-2, czwartek 15:10-16:40

1. Na zajęciach skupiliśmy się na wykonaniu programu „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.

Kod wraz z dokładnym komentarzem:

```
.olist
#include "a16def.inc"
.list
.listac
.device ATmega16
.cseg
.org 0x0000
jmp START
;wektory przerwan
.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

;Uwaga! Diody zapalamy zerem
MAIN:
ldi R17, 0b11100111 ;basowe początkowe ustawienie diód (2 - pośrodku)
out PORTB, R17 ;zapalenie ich
call delay_1s ;utrzymanie przez sekunde
;liczniki pętli
ldi R19, 3
ldi R21, 4

;rozszerzanie do zewnątrz
ROZSZ:
ldi R20, 0b11111111 ;dodatkowy rejestr, służy do utrzymania prawidłowej wartości C
MOV R18, R17 ;kopia rejestru (wykonamy 2 operacje ROL i ROR - potrzebujemy kopie)
ROL R20 ;ustawienie flagi carry na 1 przez przesunięcie oddzielnego rejestru dla prawidłowego działania ROL R17 ;rozszerzanie w celu zapalenia zewnętrznych diód
ROL R17 ;przesuwamy w lewo, z prawej wsuwamy C=1 by nie zapalać niechcianych diód, wysuwamy 1 do carry
ROR R18 ;dalej przesuwamy w prawo, wsuwając jedynkę z carry
AND R17, R18 ;operacja AND na R17 i R18 pozwala uzyskać prawidłową wartość maski, zapalającej diody które chcemy
out PORTB, R17 ;zapalamy
call DELAY_1s ;trzymamy
dec R19 ;zmniejszamy licznik
brne ROZSZ ;albo powtarzamy gdy Z!=0, albo zaczynamy zwać
;zweźzanie do środka
ZWEZ:
ldi R20, 0b11111111 ;w carry potrzebujemy 1, żeby wyłączyć diody
ROR R20 ;C=1 używujemy poprzez przesunięcie osłau jedynkę
ROL R18 ;przesuwamy w lewo (wsuwamy jedynkę z prawej), wysuwamy zero C=0
ROR R20 ;znowu ustawiamy C=1
ROR R17 ;przesuwamy w prawo (wsuwamy jedynkę z lewej)
OR R17, R18 ;wykonujemy operację OR by maska zawierała prawidłowe wartości
out PORTB, R17
call delay_1s
dec R21
brne ZWEZ
JMP MAIN ;powrót na początek

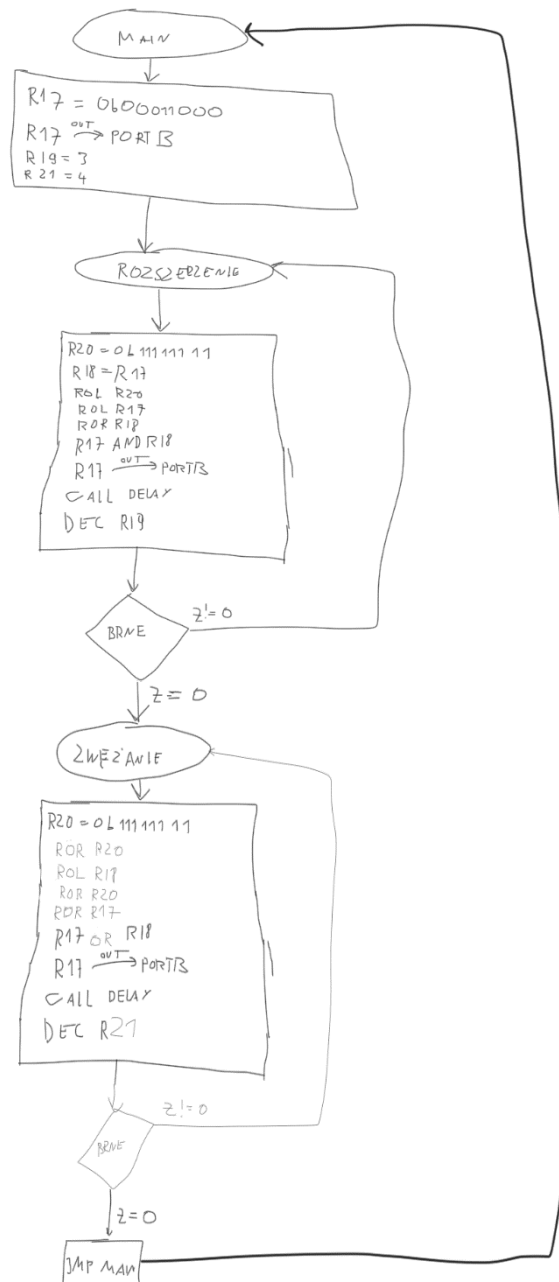
;opóźnienie 1s
DELAY_1s:
NOP
ldi R22, 21
ldi R23, 75
ldi R24, 188
OP_1s:
dec R24
brne op_1s
dec R23
brne op_1s
dec R22
brne op_1s
```

Stan początkowy i dwa kroki rozszerzania:

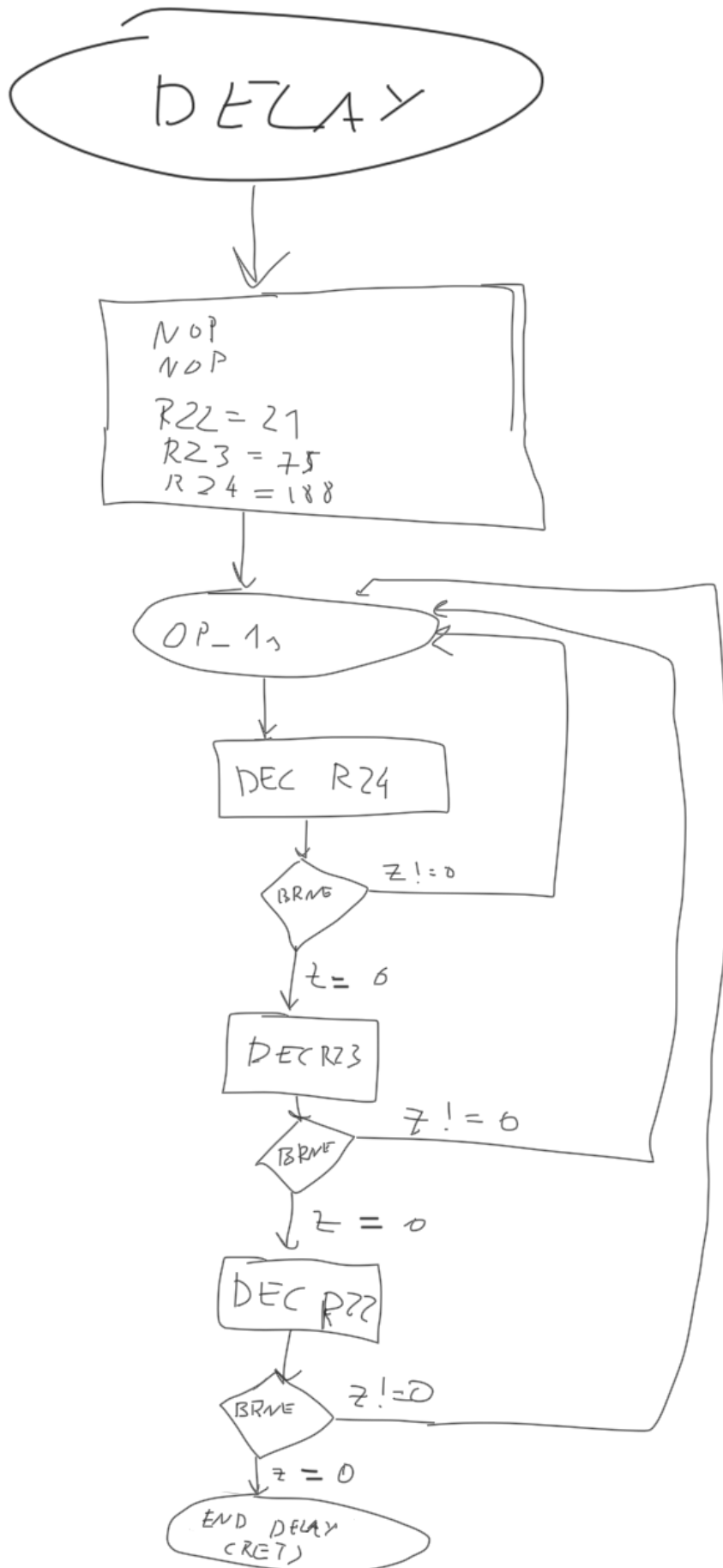
Name	Address	Value	Bits	Name	Address	Value	Bits	Name	Address	Value	Bits
DDRB	0x17 (0x37)	0xFF	11111111	DDRB	0x17 (0x37)	0xFF	11111111	DDRB	0x17 (0x37)	0xFF	11111111
PINB	0x16 (0x36)	0xE7	11100111	PINB	0x16 (0x36)	0xC3	11000011	PINB	0x16 (0x36)	0x81	10000001
PORTB	0x18 (0x38)	0xE7	11100111	PORTB	0x18 (0x38)	0xC3	11000011	PORTB	0x18 (0x38)	0x81	10000001

2. Algorytm (schemat blokowy) – pomijam przeskakiwanie przerwań i ustawianie rejestrów.

Magiczne oko:



Delay:



3. Symulacja w SimulIDE – choć program działał już na fizycznej płytce.

Szybki przeskoc jest tylko kwestią nagrania i konwersji do formatu gif, dlatego obrazek należy traktować poglądowo.

