

8. Wyświetlacz 7 segmentowy

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

Data wykonania : 15.12.2022r.

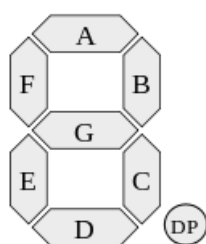
Grupa : T-2, czwartek 15:10-16:40

1. Wykorzystujemy wyświetlacz wielosegmentowy, żeby wyświetlić sekwencję 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, A, B, C, D, E, F, H, L, P. Nasz wyświetlacz ma wspólną anodę, ale przed nim jest bufor w którym korzystamy z bramki NOT, dlatego sterujemy uruchomieniem segmentów za pomocą 1.

Przykładowy wyświetlacz:



Opis segmentów:



Program:

```
.nolist
#include "m16def.inc"
.list
.listmac
.device ATmega16
.cseg
;poniżej opisane symbole - segment zapalamy '1' - stanem wysokim
.equ c_0=0b11111100
.equ c_1=0b01100000
.equ c_2=0b11011010
.equ c_3=0b11110010
.equ c_4=0b01100110
.equ c_5=0b10110110
.equ c_6=0b10111110
.equ c_7=0b11100000
.equ c_8=0b11111110
.equ c_9=0b11110110
.equ c_A=0b11101110
.equ c_B=0b00111110
.equ c_C=0b10011100
.equ c_D=0b01111010
.equ c_E=0b10011110
.equ c_F=0b10001110
.equ c_H=0b01101110
.equ c_L=0b00011100
.equ c_P=0b11001110
.org 0x0000
jmp START
;wektory przerwań
.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 ;do rejestru R16 wpisujemy same '1'
out DDRB, r16 ;za pomocą tego rejestru ustawiamy port B jako wyjście

MAIN:
ldi r17, c_0 ;zapisujemy w rejestrze R17 wartość którą chcemy wyświetlić
out portB, r17 ;wysyłamy wartość z rejestru R17 na port B, wyświetlamy ją na wyświetlaczu ('0')
call delay_1s ;sekunda opóźnienia, żeby zobaczyć wynik
;dalej wykonujemy to samo dla wszystkich pozostałych wartości z sekwencji
ldi r17, c_1
out portB, r17
call delay_1s
ldi r17, c_2
out portB, r17
call delay_1s
ldi r17, c_3
out portB, r17
call delay_1s
ldi r17, c_4
out portB, r17
call delay_1s
ldi r17, c_5
out portB, r17
call delay_1s
ldi r17, c_6
out portB, r17
```

```
call delay_1s
ldi r17, c_7
out portB,r17
call delay_1s
ldi r17, c_8
out portB,r17
call delay_1s
ldi r17, c_9
out portB,r17
call delay_1s
ldi r17, c_A
out portB,r17
call delay_1s
ldi r17, c_B
out portB,r17
call delay_1s
ldi r17, c_C
out portB,r17
call delay_1s
ldi r17, c_D
out portB,r17
call delay_1s
ldi r17, c_E
out portB,r17
call delay_1s
ldi r17, c_F
out portB,r17
call delay_1s
ldi r17, c_H
out portB,r17
call delay_1s
ldi r17, c_L
out portB,r17
call delay_1s
ldi r17, c_P
out portB,r17
call delay_1s
jmp MAIN ;wracamy do etykiety MAIN i powtarzamy sekwencję
```

;opóźnienie 1 sekunda

DELAY_1s:

NOP

NOP

ldi R22, 21

ldi R23, 75

ldi R24, 188

OP_1s:

dec R24

brne op_1s

dec R23

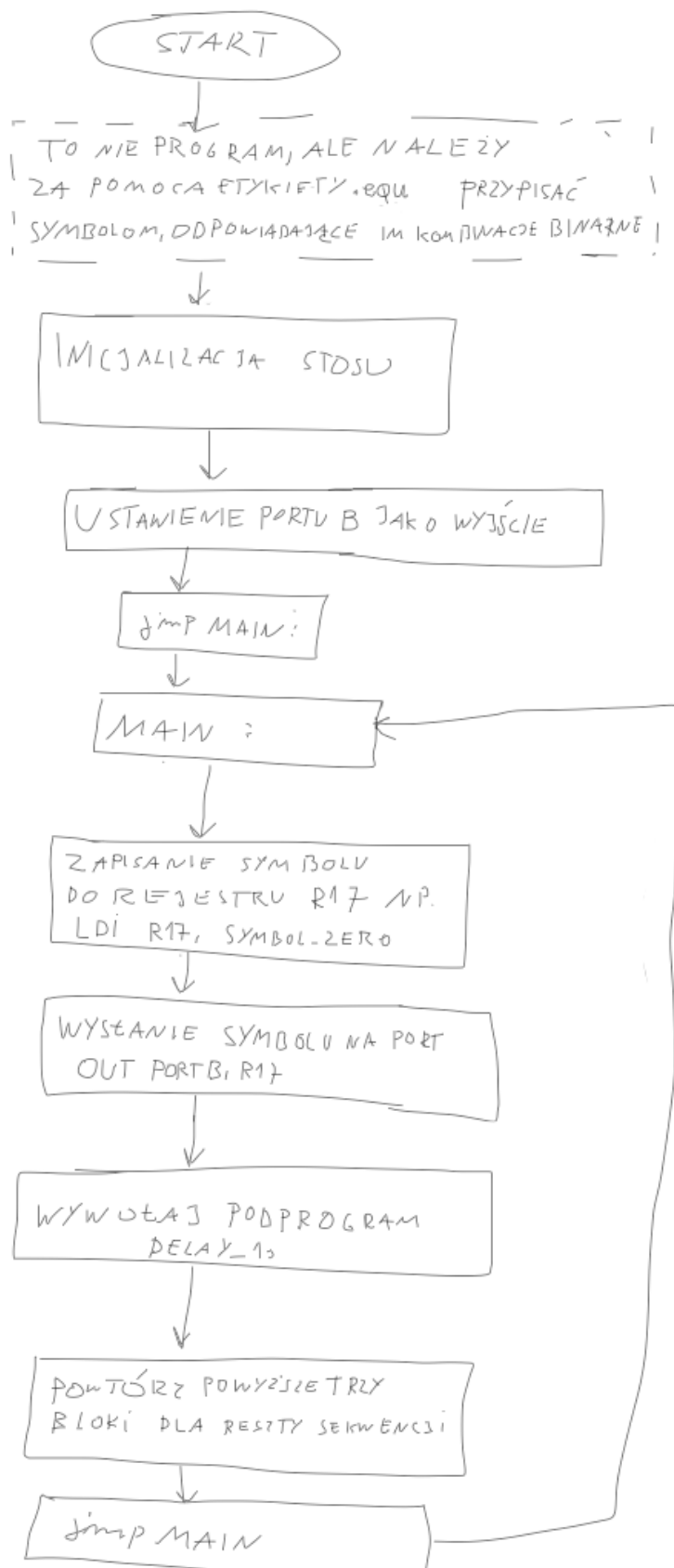
brne op_1s

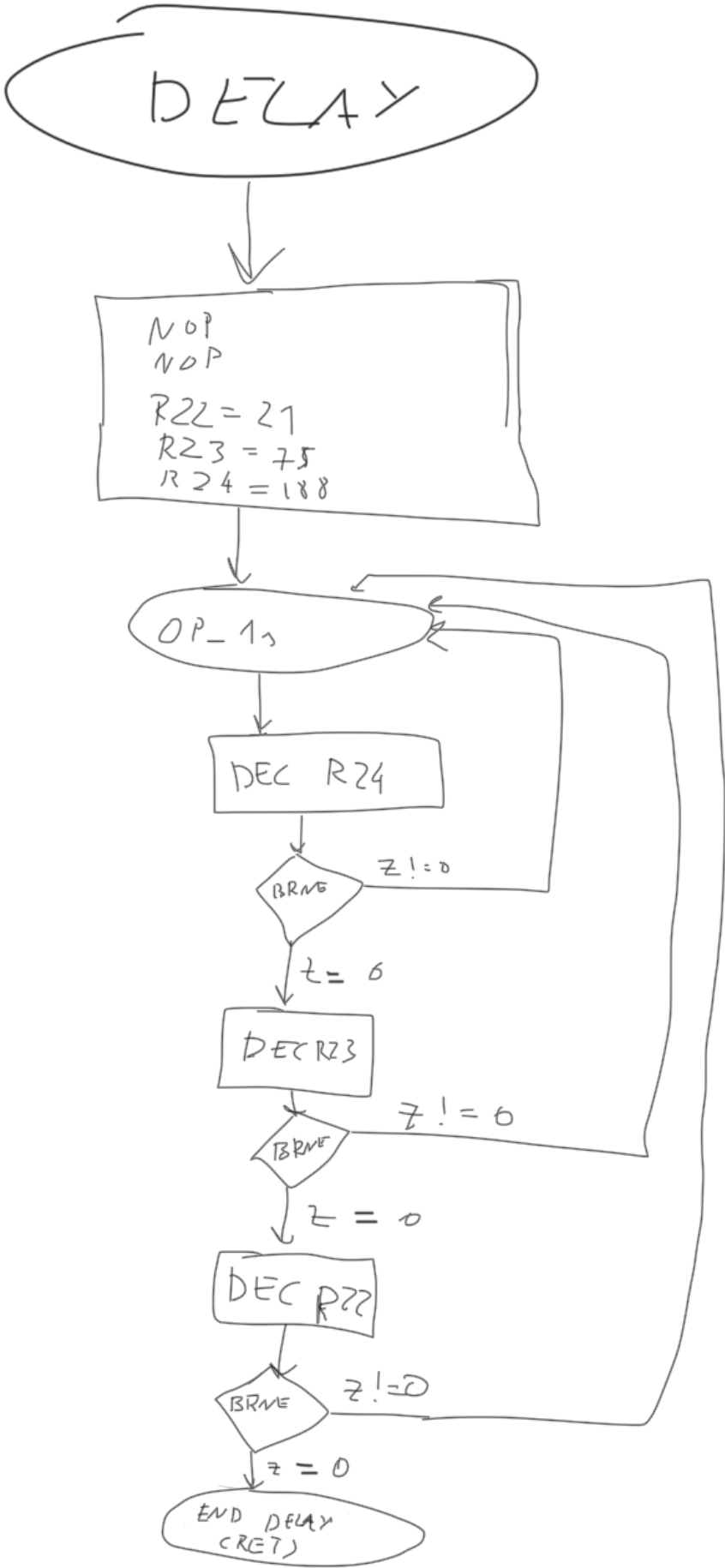
dec R22

brne op_1s

ret

Schemat blokowy – Program główny, wyświetlanie sekwencji na wyświetlaczu.





SimulIDE:

Po za symulacją, program sprawdziliśmy też na płytce i tu i tu działa prawidłowo. Pliki symulacji załączę, jeśli będzie to możliwe.

