

Sprawozdanie nr 4

Wykorzystanie wyświetlacza siedmiosegmentowego jako licznik

Wstęp

Celem laboratoriów było takie zmodyfikowanie kodów z poprzednich zajęć, by uzyskać efekt odliczania na wyświetlaczu siedmiosegmentowym płytki ZL2MCS51. Licznik miał odliczać sekundy na dwóch pierwszych od lewej wyświetlaczach, 1/10 sekundy na kolejnym i 1/100 na ostatnim wyświetlaczu. W tym celu należało wykorzystać dwa układy czasowo-licznikowe (Timer0 i Timer1) oraz dwa przerwania. Jeden Timer miał odpowiadać za wyświetlanie cyfr, a drugi za odliczanie czasu.

Podobnie jak na poprzednich zajęciach, wykorzystano mikrokontroler 8051 i zestaw uruchomieniowy ZL2MCS51, wyświetlacze 7-segmentowe do portu P0. Wykorzystano oprogramowanie MIDE-51 i Flip.

Kod programu:

```
#include "8051.h"

#define TH0_RELOAD 0xF7
#define TL0_RELOAD 0x00
#define TH1_RELOAD 0xdc // dodanie drugiego Timera (Timer1)
#define TL1_RELOAD 0x00

#define TIK 1
#define TIK1 1
int wysw = 1;

void timer0_init(void) {
    TH0 = TH0_RELOAD;
    TL0 = TL0_RELOAD;
    TMOD = TMOD | 0x01;
    TR0 = 1;
    ET0 = 1;
}

void timer1_init(void) {
    TH1 = TH1_RELOAD;
    TL1 = TL1_RELOAD;
    TMOD = (TMOD & 0xCF) | 0x10;
    TR1 = 1;
    ET1 = 1;
}

void timer_isr (void) __interrupt (1) __using (0) {
    static int count=0;
    TH0 = TH0_RELOAD;
```

```

    TL0 = TL0_RELOAD;
    count++;
    if (count==TIK) {
        count=0;
        P2_7=!P2_7;
        wysw *= 2;
        if(wysw == 16) wysw = 1;
    }
}

void timer_isr1 (void) __interrupt (3) __using (1) {
    static int count1=0;
    TH1 = TH1_RELOAD;
    TL1 = TL1_RELOAD;
    count1++;
    if (count1==TIK1) {
        count1=0;
        P2_0=!P2_0;
        num0++;
        if(num0 > 9)
        {
            num0 = 0;
            num1++;
            if(num1 >9)
            {
                num1 = 0;
                num2++;

                if(num2 > 9)
                {
                    num2 = 0;
                    num3++;

                    if(num3 > 9)
                    {
                        num3 = 0;
                    }
                }
            }
        }
    }
}

char nums[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,
0x6F};

void main(void) {

    int stateS0 = 1;
    int stateS1 = 2;
    int stateS2 = 3;

    int state = stateS0;

    int btn_current = 0xFF;
    int btn_before = 0xFF;
    int btn_zero = 0xFF;
    int time = 0;
    int maxTime = 0x0F;
    int update = 0xFF;

```

```

int skip = 0;
int tdl = 0;

EA = 0;
timer0_init();
timer1_init();

EA = 1;
P2=0xFF;
while(1) {
    P2_6 = P3_6;

    if(!skip)
    {
        if(state == stateS0)
        {
            btn_current = P3;
            if(btn_current != btn_zero)
            {
                skip = 1;
                state = stateS1;
                continue;
            }
        }
        else
        {
            if(state == stateS1)
            {
                if(btn_current != P3)
                    state = stateS2;
            }
            else{
                state = stateS0;
                update = btn_current;
                btn_current = 0xFF;
            }
        }
    }
    else{
        skip++;
        if(skip >=maxTime)
        {
            skip = 0;
        }
    }

    if(!(update & 0x01)){
        update = 0xFF;
        num0++;
        if(num0>9)
            num0=0;
    }

    if(!(update & 0x02)){
        update = 0xFF;
        num1++;
        if(num1>9)
            num1=0;
    }

    if(!(update & 0x04)){
        update = 0xFF;
        num2++;
        if(num2>9)

```

```
        num2=0;
    }

    if(update != 0xFF){
        update = 0xFF;
        num3++;
        if(num3>9)
            num3=0;
    }

    P1 = wysw;
    switch(P1){
        case 1:
            P0=nums[num0];
            break;

            case 2:
                P0=nums[num1];
                break;

            case 4:
                P0=nums[num2];
                break;

            case 8:
                P0=nums[num3];
                break;
        }
    }
}
```