

Alicja Kapiszka
Paweł Adamski
Informatyka N1, gr. 30C

Sprawozdanie nr 3

Programowa eliminacja drgań ze styków

Wstęp

Laboratoria były kontynuacją poprzednich zajęć, na których uzyskano efekt inkrementacji liczb na wyświetlaczu poprzez wciskanie odpowiednich przycisków. Poprzez zjawisko drgań zestyków, inkrementacja była jednak nieprawidłowa. Na obecnych zajęciach należało to poprawić.

Zjawisko drgań zestyków jest spowodowane tym, że po wciśnięciu przycisku stan panujący na porcie nie zmienia się od razu ze stanu wysokiego na niski, lecz występują oscylacje (kilkukrotne przechodzenie ze stanu wysokiego na niski i odwrotnie), co skutkuje wielokrotnym zwiększeniem stanu wyświetlacza. Z tego powodu konieczna jest eliminacja drgań zestyków (tzn. debouncing), co można uczynić w sposób sprzętowy lub programowy. Metody programowe, wykorzystane na tych zajęciach, polegają na odliczaniu określonego czasu (kilkadziesiąt ms) od momentu wciśnięcia przycisku i ponownym sprawdzeniu wartości portu po tym czasie. Jeśli stan ponownie będzie niski, uznaje się, że przycisk został wciśnięty. Jeśli nie – uznaje się poprzedni stan niski za skutek drgania zestyków. W implementacji takiego rozwiązania używa się funkcji opóźniającej.

Podobnie jak na poprzednich zajęciach, wykorzystano mikrokontroler 8051 i zestaw uruchomieniowy ZL2MCS51, przyciski podłączono do portu P3, a wyświetlacze 7-segmentowe do portu P0. Wykorzystano oprogramowanie MIDE-51 i Flip.

Przebieg ćwiczenia:

Należało tak zmodyfikować kod z poprzedniego ćwiczenia, określając odpowiedni czas do eliminacji drgań zestyków, aby zniwelować ich efekt.

W naszym przypadku ustawiliśmy czas na 15 ms oraz wykorzystaliśmy maszynę stanów z trzema stanami: S0 (gdy przycisk nie jest wciśnięty), S1 (gdy następuje odliczanie czasu) i S2 (oczekiwanie na zwolnienie przycisku).

```
#include "8051.h"

#define TH0_RELOAD 0xF7
#define TL0_RELOAD 0x00
#define TIK 1
int wysw = 1;

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

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

void main(void) {

    int num0 = 0;
    int num1 = 2;
    int num2 = 3;
    int num3 = 4;

    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;

    EA = 0; // zablokowanie przerwa.
    timer0_init(); // przygotowanie uk.adu Timer0
    EA = 1; // odblokowanie przerwa.
    P2=0xFF; // wygaszenie wszystkich diod
    while(1) {
        P2_6 = P3_6; // obs.uga przycisku i diody

        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;
}
}
}
}

```