POLITECHNIKA CZĘSTOCHOWSKA, **WYDZIAŁ INŻYNIERII MECHANICZNEJ I INFORMATYKI**
KATEDRA INŻYNIERII KOMPUTEROWEJ

Laboratorium Systemów Mikroprocesorowych

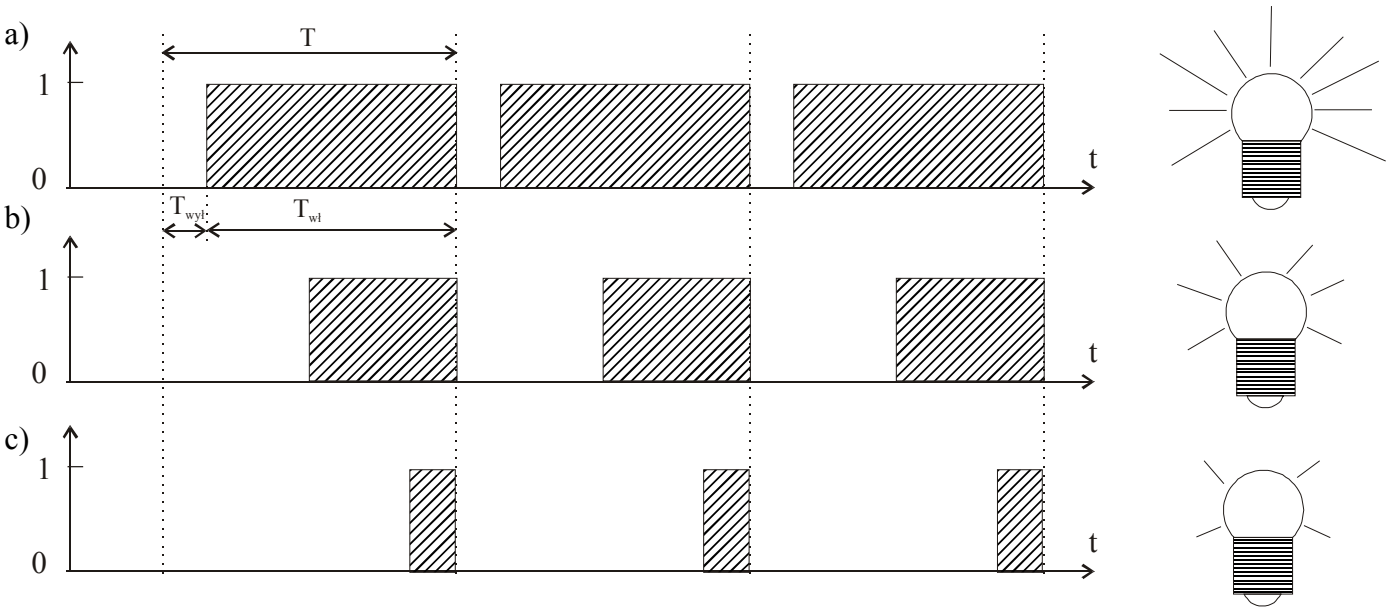
Ćwiczenie nr 4:

TEMAT: **Regulacja MSI (SAB80C537).**

1.Wiadomości podstawowe

Regulacja MSI (Modulacja Szerokości Impulsów) lub PWM (Pulse Width Modulation) polega na doprowadzeniu do odbiornika (żarówki, silnika itp.) napięcia o stałej wartości ale o zmiennym wypełnieniu (rysunek 1). Okres **T** (czyli częstotliwość również) tak powstałej fali pozostaje stały natomiast jej wypełnienie zmienia się od zera do 100%.
W przypadku sterowania tak powstałą falą żarówki, w każdym cyklu **T** jest ona przez pewien czas $\mathbf{T_{wł}}$ włączona (zakreskowane pole na rys.1) – czyli doprowadzony jest do niej prąd, oraz przez pewien czas $\mathbf{T_{wył}}$ wyłączona. Przy czym $\mathbf{T_{wył}= T-T_{wł}}$.

Jeśli czas **T** jest dostatecznie krótki (najczęściej rzędu kilkuset mikrosekund) to ze względu na bezwładności (żarówki i oka) uzyskuje się efekt płynnej regulacji jasności świecenia. W podobny sposób można uzyskać np. regulację szybkości obrotów w silnikach prądu stałego.



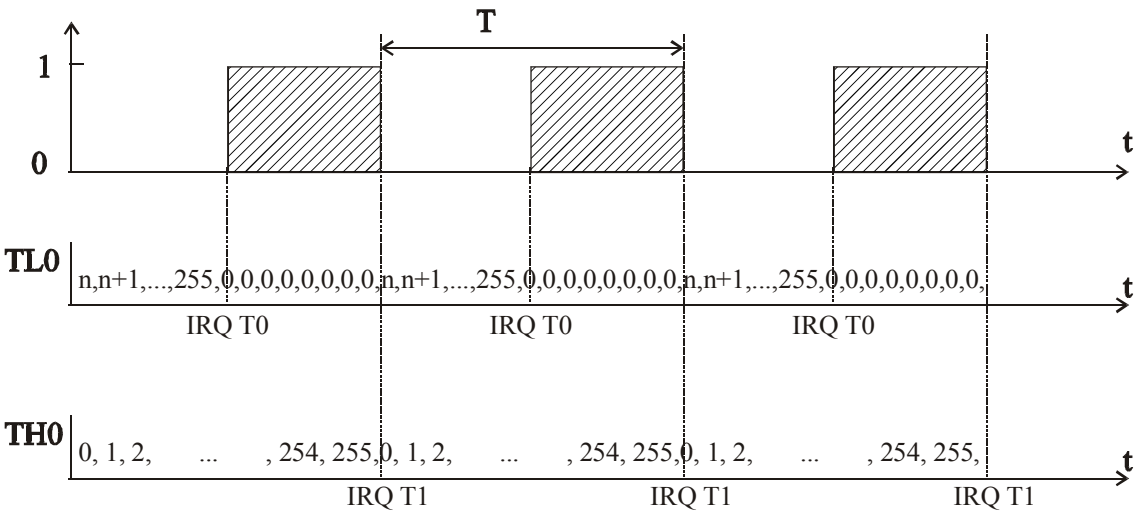
Rys.1 Wpływ wartości wypełnienia na jasność świecenia żarówki: a) wypełnienie bliskie 100% - żarówka świeci światłem b. jasnym, b) wypełnienie 50% - żarówka świeci połową swojej mocy, c) wypełnienie bliskie 0% - żarówka świeci światłem bardzo słabym.

2. Przykład

Wykorzystując technikę MSI spowodować regulację jasności świecenia wyświetlacza LED. Na wyświetlaczu powinna być wyświetlona cyfra '**7**', natomiast jasność świecenia jako liczba z zakresu 0..255 (odpowiadająca zakresowi 0..100% maksymalnej jasności świecenia) powinna być pobierana z rejestru uniwersalnego **R7**.

Rozwiązanie:

Do odliczania czasu zapalenia i zgaszenia LED wykorzystamy Timer 0 pracujący w trybie 3. Dwie części TL0 i TH0 timera T0 pracują wtedy jako dwa niezależne timery 8 bitowe. Każdy z nich generuje po swoim przepełnieniu niezależne przerwanie: TL0 – przerwanie T0, TH0 – przerwanie T1. Timer TH0 wykorzystamy do odmierzania stałego cyklu o długości T. Jego wartością początkową będzie zawsze zero a więc będzie on generował przerwanie dokładnie co 256 cykli zegarowych czyli co 256*1[us]=256[us].
Uwaga! Procedury obsługi przerwań T0 oraz T1 nie powinny być zbyt długie (najwyżej kilkanaście instrukcji). Każda instrukcja wykonywana jest ok. 1[us] i zbyt długie procedury mogłyby nie zdążyć wykonać się przed nadejściem kolejnego przerwania – co spowodowałoby prawdopodobnie „zawieszenie się" programu.



Rys.2

Timer TL0 uruchamiany będzie dokładnie w tym samym momencie co TH0 ale z wartością początkową **n** (n=0..255). Ponieważ zwiększa on swoją zawartość z taką samą szybkością jak TH0 stan przepełnienia osiągnie szybciej (dokładnie n impulsów zegarowych wcześniej).
Jeśli w momencie przepełnienia TL0 (przerwanie TF0) włączymy wyświetlacz LED, natomiast w momencie przepełnienia TH0 (przerwanie TF1) wyłączymy go (rysunek 2) to uzyskamy czas świecenia LED o długości n impulsów zegarowych w każdym cyklu **T**. Wypełnienie tak powstałych impulsów będzie więc wynosiło (n/256)*100%.
Timer TL0 po przepełnieniu powinien zostać zatrzymany do czasu przepełnienia się timera TH0 w celu zsynchronizowania początku każdego cyklu.

Fragment programu realizujący powyższe zadanie powinien wyglądać tak:

```
cyfra7        equ 11000111b        ;przykladowa kombinacja bitow zapalajaca cyfre '7' na wyswietlaczu
zgaszonyLED   equ 11111111b        ;przykladowa kombinacja bitow wylaczajaca wszystkie segmenty

      ORG 800Bh ;wektor przerwania T0
      LJMP przerwanie_T0
```

```
        ORG 801Bh ;wektor przerwania T1
        LJMP przerwanie_T1

;######## POCZATEK PROGRAMU GLOWNEGO ##############

        ORG     8100h

        lcall wlacz_timery    ;wywolaj procedure ustawiajaca timery w odpowiedni tryb i wlaczajaca je

petla:
        ;..tutaj program glowny może wykonywac swoje zadania, np. odczytywac przetwornik AC

        mov R7,#.. ;do rejestru R7 możemy wpisywac liczby z zakresu 0..255, w celu uzyskania wypelnienia
                   ;0..100%

        LJMP petla ;powtarzaj w nieskonczonej petli



;###### ustaw timery T0 i T1 w tryb 3 #######

;********* bity w rejestrze TMOD *********
;TIMER 0
T0_G    EQU     0       ;GATE
T0_C    EQU     0       ;COUNTER/-TIMER
T0_M    EQU     3       ;MODE (0..3)
TIM0    EQU     T0_M +T0_C*4 +T0_G*8

;TIMER 1
T1_G    EQU     0       ;GATE
T1_C    EQU     0       ;COUNTER/-TIMER
T1_M    EQU     3       ;MODE (0..3)
TIM1    EQU     T1_M + T1_C*4 + T1_G*8

TMOD_SET        EQU     TIM0 + TIM1*16

wlacz_timery:
        MOV     TMOD,#TMOD_SET        ;Ustawienie trybu pracy Timerow 0 i 1

        setb    TR1                   ;start Timera 1
        SETB    TR0                   ;start Timera 0

        SETB    ET0             ;wlacz zezwolenie na
        SETB    ET1             ;przerwanie od Timera 0 i Timera 1

        SETB    EAL             ;zezwolenie na przerwanie ogolne

        ret

przerwanie_T0:  ;procedura obsługi przerwania T0
        push PSW

        clr TR0 ;zatrzymaj timer TL0

        mov P6,#cyfra7 ;zapal cyfre 7 na wyswietlaczu LED

        pop PSW
        reti ;wyjdz z przerwania I wroc do programu glownego


przerwanie_T1: ;procedura obsługi przerwania T1
        push PSW

        mov P6,#zgaszonyLED ;zgas wyswietlacz LED

                   ;TH0 ma już wartosc zero (przepelnienie)
        mov TL0,R7  ;TL0 ma wartosc zalezna od tego jaka szerokosc impulsu chcemy uzyskac

        setb TR0 ;uruchom ponownie timer TL0 (zsynchronizowanie poczatku zliczania)

        ;MOV TH0,#00 ;UWAGA! patrz przypis [1)]

        pop PSW
        reti ;wyjdz z przerwania I wroc do programu glownego
```

1) Uwaga, przy wartościach wypełnienia bliskich zeru (początkowa wartość timera TL0) program może działać nieprawidłowo – generować falę o innym niż zamierzaliśmy wypełnieniu. Z powodu dość długiego czasu wykonywania pojedynczej instrukcji procesora (~1us) porównywalnego z czasem pojedynczego impulsu

podawanego na wejście timera może nastąpić niejednoczesny początek cyklu zliczania timerów TL0 i TH0. W efekcie TL0 może wygenerować swoje przerwanie później niż timer TH0, co jest sprzeczne z zaproponowaną ideą generacji MSI.
Aby temu zapobiec należy w obsłudze przerwania T1 **tuż po uruchomieniu** timera TL0 (czyli po instrukcji **SETB TR0**) wymusić stan zero w liczniku TH0 za pomocą instrukcji **MOV TH0,#00**.

Zaproponowany sposób generacji fali MSI nie jest jedynym możliwym do realizacji w sposób programowy. Do odliczania interwałów czasowych można wykorzystać te same timery (T0,T1) ale inaczej skonfigurowane, lub inne np. Timer 2. (W skrajnym przypadku można napisać program bez użycia timerów i przerwań ale niestety będzie on musiał angażować większość czasu procesora i nie można będzie poza generacją fali MSI wykonywać innych zadań). Poza tym mikrokontroler SAB 80C537 ma wbudowane sprzętowe mechanizmy do generacji MSI, umożliwiają one generację takiej fali beż jakiegokolwiek spowolnienia pracy procesora. Dzięki takiemu rozwiązaniu procesor może wykonywać wiele skomplikowanych obliczeń a wymaganą wartość współczynnika wypełnienia wpisuje do specjalnych rejestrów. Odpowiednia fala pojawia się na określonych wyprowadzeniach (portach) mikrokontrolera.

3. Program ćwiczenia

1. Napisać program (wykorzystując timery) regulujący jasność świecenia cyfr wyświetlanych na wyświetlaczu LED.. Jasność powinna być płynnie regulowana od 0 do 100% poprzez położenie gałki potencjometru. Zaobserwować wyniki. Czy efekt pulsacji przy regulacji MSI jest w tym przypadku widoczny. Czy da się w pełnym zakresie (od całkowitego zgaszenia do pełnej jasności) sterować jasnością za pomocą tak napisanego programu? Jeśli nie to dlaczego?
2. Zmodyfikować program z p.1 tak, aby cyfry wyświetlane na wyświetlaczu LED zmieniały się co 1 sekundę. Uwaga do odliczania czasu można wykorzystać procedury wykonujące wielokrotnie instrukcję NOP lub przerwanie generowane przez T0 lub T1.
3. Zmodyfikować program z p.1 tak, aby zarówno jasność cyfr jak i one same zależały od położenia gałki potencjometru. Zakres regulacji wyświetlanych cyfr powinien wynosić 0..F, przy regulacji gałki potencjometru od lewego do prawego skrajnego położenia.

4 Wnioski z ćwiczenia

Opisać sposób generacji fali MSI wykorzystany w programie. Czy jest to jedyny możliwy sposób, jeśli nie podać przykład innego. Jakie zalety i wady ma wykorzystany sposób. Czy efekt pulsacji jest widoczny na wyświetlaczu LED, wyjaśnić. Czy regulacja jasności świecenia wykonana w ten sposób pozwala procesorowi na wykonywanie innych zadań poza generacją MSI?