

Mirosław Łazoryszczak  
**LABORATORIUM nr 2**

**Temat: Obsługa wyświetlacza siedmiosegmentowego LED**

## 1. ARCHITEKTURA MCS-51 (CD.)

Do realizacji wielu zadań w systemach mikroprocesorowych niezbędny jest mechanizm odmierzania czasu względnego. Mikrokontroler 8051 jest taktowany zewnętrznym sygnałem zegarowym (ang. *clock*) i ten właśnie sygnał jest podstawowym źródłem określania czasu. W najprostszym przypadku możliwe jest zliczanie instrukcji wykonywanych przez mikroprocesor. Znając liczbę taktów zegara przypadającą na jedną instrukcję oraz częstotliwość sygnału zegarowego możliwe jest precyzyjne określenie czasu potrzebnego do wykonania sekwencyjnego fragmentu kodu programu. Oczywiście, sytuacja tak jest możliwa w przypadku programu napisanego w assemblerze. Jeśli program napisany został w języku wysokiego poziomu, to oszacowanie czasu potrzebnego na wykonanie analogicznego fragmentu programu jest znacznie trudniejsze i wymaga znajomości kompilatora lub/hoặc kodu wynikowego w języku maszynowym. Znacznie bardziej skutecznym mechanizmem odmierzania czasu jest wykorzystanie odrębnego układu czasowo-licznikowego. W celu odczytu czasu odliczanego przez układ czasowy, mikrokontroler powinien wykonywać cyklicznie odpytywanie tego pierwszego. Może to być kłopotliwe w implementacji, ponadto może stanowić przyczynę błędnych odczytów czasu, ponieważ to nie czasomierz powiadamia procesor, ale procesor sprawdza układ czasowy odczytując stan jego rejestrów. Precyzyjną obsługę procesu odmierzania czasu zapewnia mechanizm przerwań. Polega on na powiadomieniu procesora o wystąpieniu określonego zdarzenia przerywając jednocześnie działanie programu głównego. Poniżej zostaną omówione wybrane aspekty działania układu czasowo-licznikowego oraz układu przerwań.

### 1.1. UKŁAD CZASOWO-LICZNIKOWY 8051

Mikrokontroler 8051 posiada dwa układy czasowo-licznikowe nazywane TIMER 0 oraz TIMER 1. Każdy z układów może pracować w kilku trybach. Dokładny opis wszystkich trybów pracy można znaleźć w dokumentacji mikrokontrolerów rodziny 8051 [2, 4, 5]. Poniżej podana zostanie jedynie krótka charakterystyka poszczególnych trybów oraz skrócony sposób ich programowania. Układ czasowo-licznikowy może pracować w trzech trybach pracy. Poszczególne tryby są numerowane począwszy od 0. Funkcje układu w poszczególnych trybach ilustruje tabela 1.

**Tabela. 1:** Tryby pracy układu czasowo-licznikowego mikrokontrolera 8051.

Tryb	Funkcja
0	Licznik 13 bitowy
1	Licznik 16 bitowy
2	Licznik 8 bitowy z automatycznym przeładowaniem wartości
3	Dwa liczniki 8 bitowe

Źródłem impulsów dla układu czasowo-licznikowego w trybie licznika może być sygnał zewnętrzny, natomiast w trybie czasomierza (timer) może być sygnał zegarowy, a ściślej  $1/12 f_{XTAL}$ , gdzie  $f_{XTAL}$  to częstotliwość sygnału taktującego. Zawartość licznika układu timera zwiększa się raz na 12 taktów zegara, zatem w praktyce następuje zliczanie cykle maszynowych.

Z układem czasowym związane są rejestry znajdujące się w bloku rejestrów specjalnego przeznaczenia (ang. *Special Function Register, SFR*). Do rejestrów tych należą:

1. TMOD – rejestr trybu pracy układu czasowo-licznikowego (NIE JEST ADRESOWANY BITOWO!),
2. TCON – rejestr sterujący (adresowany bitowo),
3. TH0, TL0, TH1, TL1 – rejestry licznikowe, w których ma miejsce zliczanie impulsów.

W rejestrze TMOD, którego bity i ich znaczenie przedstawiono w tabeli 2, cztery najstarsze bity służą do wyboru sposobu pracy układu TIMER 1, cztery młodsze bity – układu TIMERO. W tabeli 3 przedstawiono znaczenie bitów M1 oraz M0 odpowiadających za wybór trybu pracy układu czasowo-licznikowego.

**Tabela. 2:** Rejestr TMOD wyboru trybu pracy układu czasowo-licznikowego.

GATE	C/~T	M1	M0	GATE	C/~T	M1	M0
TIMER 1				TIMER 0			
GATE	0 – układ pracuje, gdy bit TRx =1 (sterowanie programowe) 1 – jeśli bit TRx=1, układ pracuje, gdy na wejściu INTx podano stan wysoki (sterowanie sprzętowe)						
C/~T	Wybór trybu czasomierz/licznik: 0 – Timer – źródłem zliczanych impulsów jest sygnał zegarowy, 1 – Counter – źródłem zliczanych impulsów jest wejście Tx.						
M1	Bit wyboru trybu (Tabela 3)						
M0	Bit wyboru trybu (Tabela 3)						

**Tabela. 3:** Znaczenie bitów M1, M0 w rejestrze TMOD.

M1	M0	Tr0yb	Opis
0	0	0	13 bitowy Timer
0	1	1	16 bitowy Timer/Counter
1	0	2	8 bitowy Timer/Counter z automatycznym przeładowaniem wartości
1	1	3	(Timer 0) TLO – 8 bitowy Timer/Counter sterowany za pomocą bitów sterujący TIMER 0, TH0 – 8 bitowy Timer sterowany za pomocą bitów sterujący TIMER 1 (Timer 1) Timer/Counter zatrzymany

Do sterowania pracą układów czasowo-licznikowych służy rejestr TCON, którego poszczególne bity są osiągalne w sposób bezpośredni. Budowę rejestru TCON przedstawiono w tabeli 4.

**Tabela. 4:** Rejestr TCON sterujący pracą układów czasowo-licznikowych.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TF1	TCON.7	Znacznik przepełnienia układu TIMER 1. Ustawiany sprzętowo. Kasowany sprzętowo po rozpoczęciu przez procesor procedury obsługi przerwania					
TR1	TCON.6	Programowe włączanie (1) lub wyłączenie (0) układu czasowo-licznikowego nr 1					
TF0	TCON.5	Znacznik przepełnienia układu TIMER 0. Ustawiany sprzętowo. Kasowany sprzętowo po rozpoczęciu przez procesor procedury obsługi przerwania					
TR0	TCON.4	Programowe włączanie (1) lub wyłączenie (0) układu czasowo-licznikowego nr 0					
IE1	TCON.3	Znacznik wystąpienia aktywnego zbocza sygnału na linii INT1. Ustawiany i kasowany sprzętowo.					
IT1	TCON.2	Bit sterujący rodzaju wyzwalania na linii INT1. Ustawiany i kasowany programowo.					
IE0	TCON.1	Znacznik wystąpienia aktywnego zbocza sygnału na linii INT0. Ustawiany i kasowany sprzętowo.					
IT0	TCON.0	Bit sterujący rodzaju wyzwalania na linii INT0. Ustawiany i kasowany programowo.					

Bieżący stan licznika nr 1 dostępny jest z pomocą rejestrów TH1, TL1, zaś bieżący stan licznika nr 0 dostępny jest za pomocą rejestrów TH0 oraz TLO. Zawartość tych rejestrów zmienia się automatycznie zgodnie z wybranym trybem pracy, może być także ustawiana w sposób programowy. Np. w trybie pracy nr 2 (licznik 16 bitowy) stan licznika nr 0 odzwierciedla para rejestrów TH0 (bardziej znaczący bajt) oraz TLO (mniej znaczący bajt). Jeśli mikrokontroler jest skonfigurowany w taki sposób, aby przepełnienie łącznie traktowanej pary rejestrów TH0|TLO zgłaszało wystąpienie przerwania, to można skrócić okres pomiędzy dwoma kolejnymi przepełnieniami przez programowe ustawienie zawartości liczników TH0|TLO.

## 1.2. KONTROLER PRZERWAŃ

Mikrokontroler 8051 wyposażony jest kontroler przerwań. Źródłem przerwań mogą być sygnały zewnętrzne lub zdarzenia wewnętrzne. Zatem przerwania mogą być wywoływane przez linie zewnętrzne INT0 oraz INT1 – odpowiadają im bity IE0 i IE1 z rejestru TCON, mogą być zgłaszane przez liczniki wewnętrzne za pomocą bitów IF0 oraz IF1. Źródłem przerwania może być także układ transmisji szeregowej (bity RI oraz TI określają precyzyjniej źródło przerwania) a także układ czasowo-licznikowy nr 2 (układ 8052).

W przypadku kontrolerów rodziny MCS-51 z każdym przerwaniem związany jest adres procedury obsługi przerwania. Adresy te są na stałe przypisane lokalizacjom w pamięci programu. Poszczególne adresy początków procedur obsługi przerwań wyszczególniono w tabeli nr 5.

**Tabela. 5:** Adresy procedur obsługi przerwań.

Numer przerwania	Źródło przerwania	Adres
0	IE0	0003h
1	TF0	000Bh
2	IE1	0013h
3	TF1	001Bh
4	RI   TI	0023h
5	TF2   EXF2	002Bh

Przerwania w mikrokontrolerze 8051 mogą być blokowane w całości lub oddzielnie za pośrednictwem rejestru IE. Znaczenie poszczególnych bitów tego rejestru przedstawia Tabela 6.

**Tabela. 6:** Rejestr IE blokowania przerwań.

EA	-	ET2	ES	ET1	EX1	ET0	EX0
EA	IE.7	0 – blokowanie wszystkich przerwań, 1 – indywidualna blokada przerwań za pomocą poniższych bitów					
-	IE.6	zarezerwowany					
ET2	IE.5	Blokada lub zezwolenie na przerwanie przepełnienia licznika nr 2 (8052)					
ES	IE.4	Blokada lub zezwolenie na przerwanie od układu transmisji szeregowej					
ET1	IE.3	Blokada lub zezwolenie na przerwanie przepełnienia licznika nr 1					
EX1	IE.2	Blokada lub zezwolenie na przerwanie na linii zewnętrznej INT1					
ET0	IE.1	Blokada lub zezwolenie na przerwanie przepełnienia licznika nr 0					
EX0	IE.0	Blokada lub zezwolenie na przerwanie na linii zewnętrznej INT0					

Kontroler przerwań 8051 jest układem priorytetowym. Oznacza to, że przerwanie o wyższym priorytecie nie może zostać przerwane przez przerwanie o niższym priorytecie. Odwrotna sytuacja jest możliwa, tzn. przerwanie o niższym priorytecie może zostać przerwane przez przerwanie o wyższym priorytecie. Dodatkowo w mikrokontrolerze wyróżnione są dwa poziomy przerwań. Przerwanie poziomu 1 posiada wyższy priorytet niż przerwanie poziomu 0. W przypadku, gdy wszystkie przerwania posiadają taki sam poziom decyduje ustalona przez producenta kolejność obsługi przerwań. Kolejność obsługi przerwań (od najwyższego priorytetu do najniższego) w ramach poziomu przerwań jest następująca:

- IE0,
- TF0,
- IE1,
- TF1,
- RI lub TI,
- TF2 lub EXF2.

Poziomy przerwań mogą zostać zdefiniowane w rejestrze IP (Tabela 7), poprzez przypisanie poszczególnym przerwanom wartości 0 lub 1.

**Tabela. 7:** Rejestr TCON sterujący pracą układów czasowo-licznikowych.

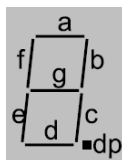
-	-	PT2	PS	PT1	PX1	PT0	PX0
-	IP.7	zarezerwowany					
-	IP.6	zarezerwowany					
PT2	IP.5	Timer 2 (8052)					
PS	IP.4	Układ transmisji szeregowej					
PT1	IP.3	Timer 1					
PX1	IP.2	Przerwanie zewnętrzne INT1					
PT0	IP.1	Timer 0					
PX0	IP.0	Przerwanie zewnętrzne INTO					

## 2. ŚRODOWISKO PRACY – ELEMENTY DODATKOWE

Do wykonania ćwiczeń laboratoryjnych opartych na mikrokontrolerze 8051 wykorzystany zostanie zestaw uruchomieniowy ZL2MCS51 [1] oraz bezpłatne oprogramowanie obejmujące kompilator języka C oraz programator. Poniżej omówione zostaną wybrane aspekty budowy płyty uruchomieniowej niezbędne do realizacji zadań w zakresie budowy wyświetlaczy siedmiosegmentowych oraz wybrane aspekty ich oprogramowania.

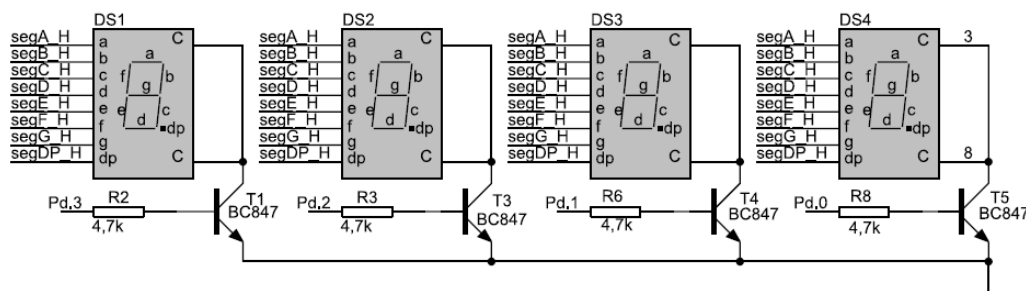
### 2.1. ZESTAW WYŚWIELACZY LED NA PŁYTCIE ZL2MCS51

Na płycie uruchomieniowej ZL2MCS51 znajdują się cztery wyświetlacze siedmiosegmentowe, służące do prezentacji danych, przede wszystkim w postaci liczbowej. Pojedynczy wyświetlacz siedmiosegmentowy składa się z siedmiu diod LED umieszczonych we wspólnej obudowie w taki sposób, że tworzą cyfrę „8”. Każda z diod w pojedynczym wyświetlaczu posiada oznaczenie literowe: *a*, *b*, *c*, *d*, *e*, *f*, *g* (rys. 1). Dodatkowo na wyświetlaczu znajduje się dodatkowa dioda, oznaczona jako *dp*, pełniąca funkcję kropki dziesiętnej.



Rys. 1. Oznaczenia segmentów wyświetlacza LED.

W zestawie ZL2MCS51 zastosowano wyświetlacze ze wspólną katodą, co oznacza, że aby zaświecić wybraną diodę w wyświetlaczu siedmiosegmentowym należy doprowadzić niski poziom napięcia, zaś do drugiego bieguna (anody) poziom wysoki. Gdyby wyświetlacze pracowały w sposób niezależny, należałoby wykorzystać do ich sterowania 4x8=32 linie (włączając kropkę dziesiętną) oraz jedną wspólną (lub cztery oddzielne) linie sterowania katodami. Taki sposób wyświetlania wymagałby użycia większej liczby portów, niż oferuje mikrokontroler. Alternatywę stanowi wyświetlanie multipleksowane lub sekwencyjne.



Rys. 2. Połączenie wyświetlaczy siedmiosegmentowych na płycie ZL2MCS51

W zestawie ZL2MCS51 wyprowadzenia diod odpowiednich segmentów dla poszczególnych wyświetlaczy są ze sobą połączone (rys. 2). Wskazują na to takie same etykiety sygnałów z lewej strony wyświetlaczy. Wynika z tego, iż podanie dowolnej kombinacji bitów na wyprowadzenia segA\_H ÷ segDP\_H skutkuje wyświetleniem takiego samego wzorca na wszystkich wyświetlaczach jednocześnie, przy założeniu, że do wszystkich zostanie jednocześnie doprowadzone odpowiednie napięcie katody.

Pożądanym jednak rozwiązaniem jest możliwość wyświetlania różnych znaków na różnych wyświetlaczach. W tym celu wykorzystuje się niedoskonałość ludzkiego oka, dzięki której ruchome obrazy, które zmieniają się z częstością większą niż ok. 25 razy na sekundę stwarzają wrażenie płynnych zmian, a ewentualne krótkie przerwy pomiędzy nimi stają się niezauważalne. Korzystając z tego faktu można zaproponować taki sposób wyświetlania, który polega na cyklicznej (sekwencyjnej) zmianie cyfr przekazywanych na poszczególne wyświetlacze z jednoczesną zmianą sterowania katody.

Dla przykładu wyświetlenie np. liczby 1234 odbywa się w sposób następujący:

- Na wyświetlacz DS1 doprowadzamy kombinację bitów odpowiadającą za wyświetlenie cyfry 1 (segmenty b i c),
- Zapewniamy odpowiedni poziom sygnału na wejściu katody wyświetlacza DS1, umożliwiającą zaświecenie się wybranych diod, dbając jednocześnie o to, aby „wygasić” pozostałe wyświetlacze (DS2, DS3, DS4),
- Na wyświetlacz DS 2 doprowadzamy kombinację bitów odpowiadającą za wyświetlenie cyfry 2 (segmenty a, b, g, e, d),
- Zapewniamy odpowiedni poziom sygnału na wejściu katody wyświetlacza DS2, umożliwiającą zaświecenie się wybranych diod, dbając jednocześnie o to, aby „wygasić” pozostałe wyświetlacze (DS1, DS3, DS4),
- Powtarzamy analogicznie operacje dla wyświetlacza DS3 (cyfra 3) i wyświetlacza DS4 (cyfra 4).

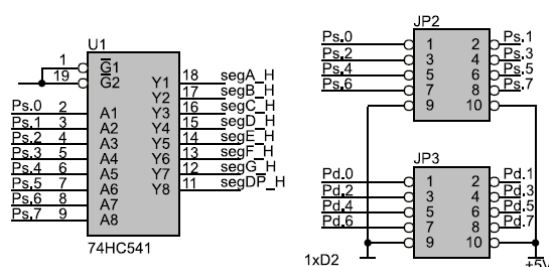
Wyświetlenie cyfr dziesiętnych (możliwe są również symbole szesnastkowe) należy zaprojektować rodzaj programowego enkodera kodu dziesiętnego na kod wyświetlacza siedmiosegmentowego. Tabela 8 zawiera możliwe kombinacje segmentów odpowiadające symbolom szesnastkowym.

**Tabela. 8:** Kod wyświetlacza siedmiosegmentowego.

Cyfra	Segmenty wyświetlacza								Kod hex
	dp	g	f	e	d	c	b	a	
0	0	0	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5B
3	0	1	0	0	1	1	1	1	0x 4F
4	0	1	1	0	0	1	1	0	0x 66
5	0	1	1	0	1	1	0	1	0x 6D
6	0	1	1	1	1	1	0	1	0x 7D
7	0	0	0	0	0	1	1	1	0x 07
8	0	1	1	1	1	1	1	1	0x 7F
9	0	1	1	0	1	1	1	1	0x 6F
A	0	1	1	1	0	1	1	1	0x 77
B	0	1	1	1	1	1	0	0	0x 7C
C	0	0	1	1	1	0	0	1	0x 39
D	0	1	0	1	1	1	1		0x 5E
E	0	1	1	1	1	0	0	1	0x 79
F	0	1	1	1	0	0	0	1	0x 71

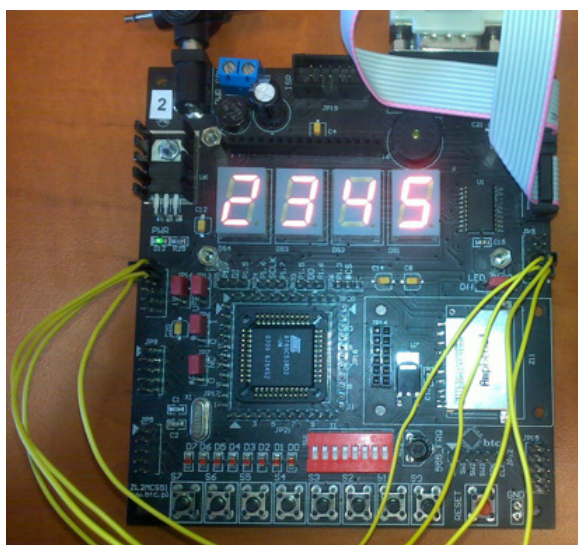
Na płycie prototypowej ZL2MCS51 połączone ze sobą wyprowadzenia segmentów doprowadzone są o układu 74HC541, który pełni funkcję bufora trójstanowego oraz wzmacniacza prądu (drivera linii). Dopiero wejścia tego układu o etykietach Ps.0. ÷ Ps.7 doprowadzone są do złącza JP2 (rys. 3). Natomiast do złącza JP3 doprowadzone są bazy tranzystorów T2 ÷ T5 poprzez rezystory R2, R3, R6, R6. Wyprowadzenia te posiadają etykiety Pd.0 ÷ Pd.3, co oznacza, że tylko cztery linie są wykorzystane w złączu JP3. Zadaniem wspomnianych wcześniej tranzystorów jest sterowanie włączeniem/wyłączeniem wspólnych katod poszczególnych

wyświetlaczy. Dodatkowo tranzystor odwraca fazę napięcia, co oznacza, iż aby włączyć dany wyświetlacz na wyprowadzeniach Pd.0 ÷ Pd.3 należy zapewnić wysoki poziom napięcia. Do zaświecenia wybranego segmentu na danym wyświetlaczu (czyli do wyprowadzeń Ps.0 ÷ Ps.7) również niezbędne jest doprowadzenia wysokiego poziomu napięcia.



Rys. 3. Układ bufora 74HC541 oraz złącza JP2 oraz JP3z wyprowadzeniami wyświetlaczy.

W celu zapewnienia możliwości sterowania układem wyświetlaczy przez mikrokontroler należy wykonać połączenie złączy odpowiednich linii złączy JP2 oraz JP3 z wybranymi portami mikrokontrolera. W ćwiczeniu przyjęto, że złącze JP2 zostanie połączone z portem P0 za pośrednictwem złącza JP10 (wykorzystany zostanie cały port P0), natomiast złącze JP3 zostanie połączone z portem P1 za pośrednictwem złącza JP4 (wykorzystane zostaną tylko cztery bity P1.0 ÷ P1.3). Poprawne połączenia przedstawiono na rys. 4. Połączenie złącza JP2 z JP10 zrealizowano za pomocą taśmy przewodów, zaś połączenie złącza JP3 ze złączem JP4 za pomocą oddzielnych przewodów.



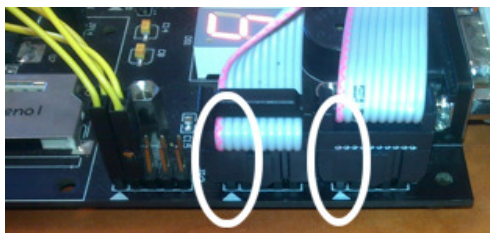
Rys. 4. Wygląd poprawnych połączeń na płycie ZL2MCS51 do obsługi wyświetlacza siedmiosegmentowego.

Ze względu na niebezpieczeństwo uszkodzenia układu ZL2MCS51 należy zwrócić szczególną uwagę na sposób wykonania połączeń. Przy złączach JP2, JP3, JP4, JP10 (a także JP5 i JP9) symbolem białego trójkąta zaznaczono wyprowadzenie nr 1. Również taśma połączeniowa posiada wyróżniony kolorem czerwonym przewód nr 1. Należy zatem zadbać, aby po obu stronach taśmy przewód nr 1 został połączony z wyprowadzeniem nr 1 w odpowiednim złączu (rys. 5).



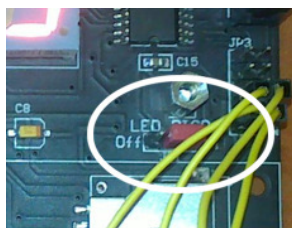
#### UWAGA:

Należy zwrócić szczególną uwagę na sposób połączenia złączy. Z uwagi na obecność linii zasilania oraz masy w wykorzystywanych złączach istnieje niebezpieczeństwo USZKODZENIA układu! Poprawność połączeń powinna zostać potwierdzona przez prowadzącego zajęcia przed podłączeniem układu do zasilania.



Rys. 5. Sposób połączenie złączy JP2 oraz JP10 za pomocą taśmy przewodowej.

W podobny sposób wykonujemy połączenia za pomocą oddzielnych przewodów pomiędzy złączami JP3 oraz JP4, tzn. wyprowadzenie nr 1 (oznaczone trójkątem) złącza JP3 łączymy z wyprowadzeniem nr 1 złącza JP4, wyprowadzenie nr 2 złącza JP3 łączymy z wyprowadzeniem nr 2 złącza JP4 itd.

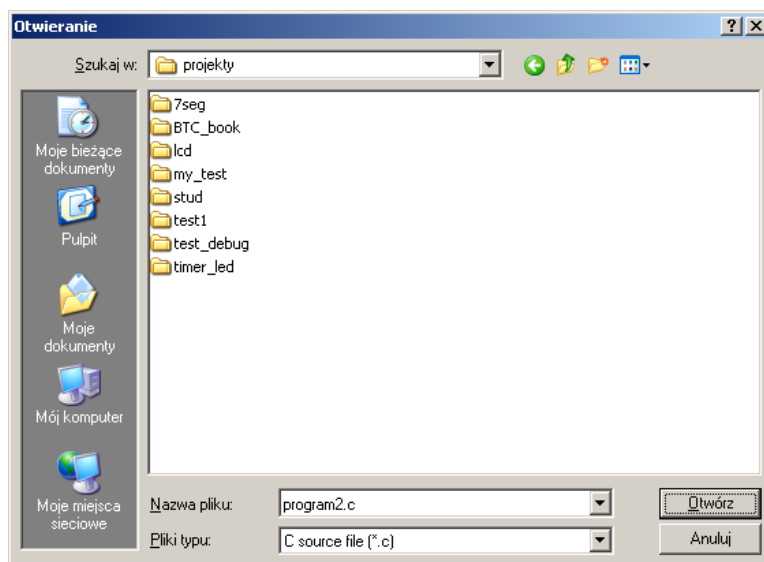


Rys. 6. Położenie zworki JP1 uaktywniającej wyświetlacz siedmiosegmentowy.

Dodatkowo należy zwrócić uwagę na położenie zworki uaktywniającej wyświetlacz siedmiosegmentowy (JP1, LED\_DISP). Powinna się ona znajdować w położeniu „On”.

## 2.2. ŚRODOWISKO PROGRAMOWE

Programy do obsługi wyświetlaczy siedmiosegmentowych w ramach Laboratorium nr 2 zostaną napisane w języku C. Kompilator języka C na mikrokontrolery 8051 dostępny jest w środowisku MIDE-51. Przy tworzeniu programu obowiązują takie same zasady jak w przypadku programów w asemblerze [3], z zastrzeżeniem, że rozszerzenie pliku z programem („.C”) powinno zostać nadane ręcznie łącznie z wyborem typu programu (rys. 6).



Rys. 7. Nadanie nazwy i wybór typu dla programu w języku C.

Na wydruku 1 przedstawiono podstawową strukturę programu w języku C na mikrokontroler 8051. W linii nr 1 następuje dołączenie standardowego pliku nagłówkowego zawierającego definicje m. in. rejestrów specjalnych i roboczych, dzięki czemu możliwe jest bezpośrednie użycie symbolicznych nazw rejestrów zamiast ich adresów.

```
01  #include <8051.h>
02
03  void main(void) {
04      while(1) {
05      }
06  }
```

Wydruk 1: Elementarny program na mikrokontroler 8051 w języku C zawierający nieskończoną, pustą pętlę główną

Wydruk nr 2 przedstawia program podstawowej obsługi klawiszy (przycisków) oraz diod LED (analogiczny do programu z Laboratorium nr 1) w języku C. W linii 05 ma miejsce przypisanie portowi P2 wartości portu P3 (dotyczy wszystkich ośmiu bitów portu).

```
01  #include <8051.h>
02
03  void main(void) {
04      while(1) {
05          P2 = P3;    // Przypisanie wartości portom
06      }
07  }
```

Wydruk 2: Program na mikrokontroler 8051 z elementarną obsługą klawiszy

Możliwe jest także operowanie na poszczególnych bitach portów, a także rejestrów specjalnych, które mają możliwość adresowania bitowego. Przykład podano na wydruku 3.

```
01  #include <8051.h>
02
03  void main(void) {
04      while(1) {
05          P2.0 = P3.0; // Przypisanie wartości bitom portu
06      }
07  }
```

Wydruk 3: Program na mikrokontroler 8051 z elementarną obsługą pojedynczego klawisza i diody

W niniejszym ćwiczeniu niezbędne będzie zastosowanie procedury obsługi przerwania od układu czasowo-licznikowego. Definicję procedury obsługi przerwania przedstawiono na wydruku 4. Składnia tej definicji jest właściwością kompilatora języka C dołączonego w środowisku MIDE-51. Inne kompilatory mogą używać nieco innej definicji.

```
01  void nazwa_procedury (void) __interrupt (1) __using (0) {
02  }
```

Wydruk 4: Program na mikrokontroler 8051 z elementarną obsługą pojedynczego klawisza i diody

Wartość w nawiasie po słowie kluczowym `__interrupt` oznacza numer procedury obsługi przerwania zgodny z tabelą nr 5. Z kolei numer przerwania związany jest z konkretnym źródłem przerwania. Należy zwrócić uwagę, iż procedura obsługi przerwania jest realizowana automatycznie przez mikrokontroler w przypadku zajścia odpowiedniego zdarzenia oraz odpowiedniej konfiguracji kontrolera przerw (np. dopuszczenie przerw). W omawianym przykładzie jest to procedura obsługi przerwania od układu czasowego TIMER 0. Wartość w nawiasie po słowie kluczowym `__using` oznacza numer bloku rejestrów, który powinien być wykorzystywany w procedurze obsługi przerwania. Do dyspozycji znajdują się bloki o numerach 0 ÷ 3 [3].



### 3. ZADANIA

W celu zapoznania się z konfiguracją i wykorzystaniem układu czasowo-licznikowego, a także z mechanizmem obsługi przerwań podany zostanie poniżej przykład programu służącego do „migania” diodą w podobny sposób jak miało to miejsce w przypadku Laboratorium nr 1. Przykładowy program przedstawia wydruk 5.

```
01  #include "8051.h" // zbiór definiujący rejestry procesora
02
03  #define TH0_RELOAD 0x00
04  #define TL0_RELOAD 0x00
05  #define TIK 1
06
07  void timer0_init(void) {
08      TH0 = TH0_RELOAD; // załadowanie czasu odliczania
09      TL0 = TL0_RELOAD; // TH0 - starszy bajt, TL0 - młodszy
10      TMOD = TMOD | 0x01; // tryb nr 1 układu TIMER 0
11      TR0 = 1; // TIMER 0 start
12      ET0 = 1; // odblokowanie przerwań od TIMER 0
13  }
14
15  void timer_isr (void) __interrupt (1) __using (0) {
16      static int count=0;
17      TH0 = TH0_RELOAD; // załadowanie czasu odliczania
18      TL0 = TL0_RELOAD;
19      count++;
20      if (count==TIK) { // dodatkowy licznik przerwań
21          count=0; // umożliwiający uzyskanie dodat.
22          P2_7=!P2_7; // opóźnień
23      }
24  }
25
26  void main(void) {
27      EA = 0; // zablokowanie przerwań
28      timer0_init(); // przygotowanie układu Timer0
29      EA = 1; // odblokowanie przerwań
30      P2=0xFF; // wygaszenie wszystkich diod
31      while(1) {
32          P2_6 = P3_6; // obsługa przycisku i diody
33      }
34  }
```

Wydruk 5: Program na mikrokontroler 8051 z wykorzystaniem układu czasowo-licznikowego i procedury obsługi przerwania

#### 3.1. ZADANIE NR 1

1. Przeanalizuj kod z wydruku 5 i określ jego funkcje.
2. Zaimplementuj program przedstawiony na wydruku 5
3. Dokonaj zmiany stałych TH0\_RELOAD i TL0\_RELOAD, a następnie stałej TIK i zaobserwuj wpływ zmian na częstotliwość migania diody.

#### 3.2. ZADANIE NR 2

1. Wzorując się na programie z wydruku 5 oraz na niniejszej instrukcji do ćwiczeń napisz program, którego zadaniem będzie jednoczesne wyświetlenie różnych cyfr, np zdefiniowanych na stałe w programie, na wyświetlaczu siedmiosegmentowym. Dobierz częstotliwość układu czasowo-licznikowego tak, aby uzyskać minimalne migotanie wyświetlaczy.

2. Rozbuduj program z punktu 1 dodając obsługę klawiszy w ten sposób, że wciśnięcie klawisza np. S0 spowoduje zwiększenie wartości wyświetlanej na wyświetlaczu DS1, S1 na wyświetlaczu DS2 itd. Po przepełnieniu wyświetlacza, czyli np. po osiągnięciu wartości „9”, kolejne wciśnięcia klawisza powinno spowodować rozpoczęcie zliczania od wartości „0”.
3. Zaobserwuj reakcję wyświetlaczy na wciśnięcie klawiszy i spróbuj wyjaśnić przyczyny tego zjawiska.

**WSKAZÓWKI:**

- Procedura obsługi przerwania powinna zawierać absolutnie niezbędne instrukcje. Wszelka obsługa np. wyświetlania powinna zostać zaimplementowana w programie głównym, tak, aby procedura obsługi przerwania zajmowała możliwie najmniej czasu procesora.
- Należy zwrócić uwagę, aby wartości początkowe liczników TH0 i TL0 (łącznie) nie powodowały zbyt częstego generowania przerw, gdyż wówczas procesor nie będzie w stanie realizować zadań w pętli głównej.
- Czas wyświetlania (doprowadzenia właściwego sygnału do katody) powinien być równomiernie rozłożony pomiędzy wszystkie wyświetlacze, w przeciwnym razie widoczne będą różnice w intensywności wyświetlania poszczególnych wyświetlaczy.

---

**4. LITERATURA**

---

- [1] BTC Korporacja: Dokumentacja techniczna zestawu uruchomieniowego ZL2MCS51, <http://www.btc.pl/pdf/zl2mcs51.pdf> (dostęp: luty 2011).
- [2] Intel, MCS51 Microcontroller Family User's Manual, 1994 (dokumentacja dostępna pod różnymi adresami w sieci Internet).
- [3] Systemy wbudowane - Laboratorium nr 1, instrukcja do ćwiczeń.
- [4] Majewski J.: Programowanie mikrokontrolerów 8051 w języku C. Pierwsze kroki, BTC, Warszawa, 2005.
- [5] Stępień C. (red.): Mikroprocesory firmy Intel, PWN, Warszawa 1992.