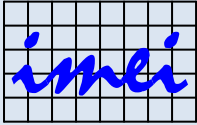


PODSTAWY TECHNIKI MIKROPROCESOROWEJ Laboratorium Elektrotechnika, studia stacjonarne pierwszego stopnia	 Instytut Metrologii, Elektroniki i Informatyki
Temat: Obsługa 4-modułowego wyświetlacza LED w sposób multipleksowany	
Opracowanie instrukcji: dr inż. Mirosław Kozioł	Ćwiczenie 6

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z multipleksowaną obsługą wielomodułowego, 7-segmentowego wyświetlacza LED oraz obsługą przerwań w mikrokontrolerach rodziny MCS-51.

Przygotowanie do ćwiczenia


Przed przystąpieniem do ćwiczenia należy przygotować odpowiedzi na poniższe pytania.

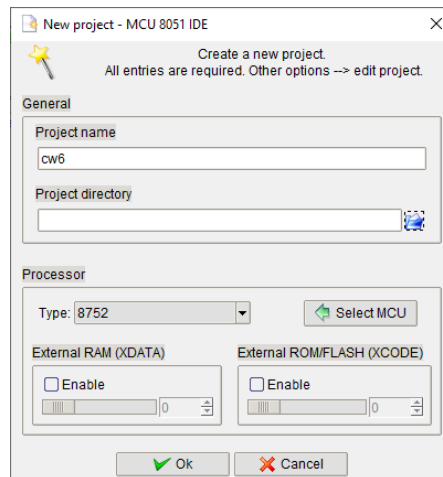
1. W jaki sposób realizowane jest wyświetlanie na zestawie 7-segmentowych wyświetlaczy LED w sposób multipleksowany?
2. W jaki sposób z poziomu języka C używając kompilatora SDCC dla mikrokontrolerów rodziny MCS-51 umieścić w pamięci kodu mikrokontrolera stałe wartości?
3. Które rejestry odpowiadają za ustawienie trybu pracy wewnętrznych jednostek czasowo-licznikowych T0 i T2 mikrokontrolerów rodziny MCS-51 i jakie jest znaczenie ich bitów?
4. Które bity odpowiadają za sterowanie pracą (m.in. strat liczników do zliczania) jednostek czasowo-licznikowych T0 i T2?
5. Który rejestr odpowiada za uaktywnienie przerwań w mikrokontrolerach rodziny MCS-51 i jakie jest znaczenie jego bitów?
6. Który rejestr odpowiada za ustalanie priorytetów przerwań w mikrokontrolerach rodziny MCS-51 i jakie jest znaczenie jego bitów?
7. W jaki sposób w języku C używając kompilatora SDCC dla mikrokontrolerów rodziny MCS-51 oznacza się funkcje, które mają obsługiwać przerwania od określonych źródeł?
8. Co jaki czas inkrementowana jest zawartość rejestrów licznika w mikrokontrolerach rodziny MCS-51 przy założeniu, że częstotliwość dołączonego do mikrokontrolera rezonatora kwarcowego wynosi 12 MHz?
9. Jaki tryb pracy licznika T2 należy wybrać i jakie nadać mu wartości początkowe, aby możliwe było odmierzenie czasu wynoszącego 1 ms, jeśli mikrokontroler taktowany jest sygnałem zegarowym o częstotliwości 12 MHz?
10. W jaki sposób wykorzystując licznik T0 odmierzyć czas 50 ms?
11. Za pomocą jakich działań arytmetycznych można określić wartość cyfry dziesiątek i jedności 8-bitowej liczby?

Program ćwiczenia

Wykorzystując środowisko MCU 8051 IDE napisz program w języku C implementując kolejno przedstawione poniżej zadania.

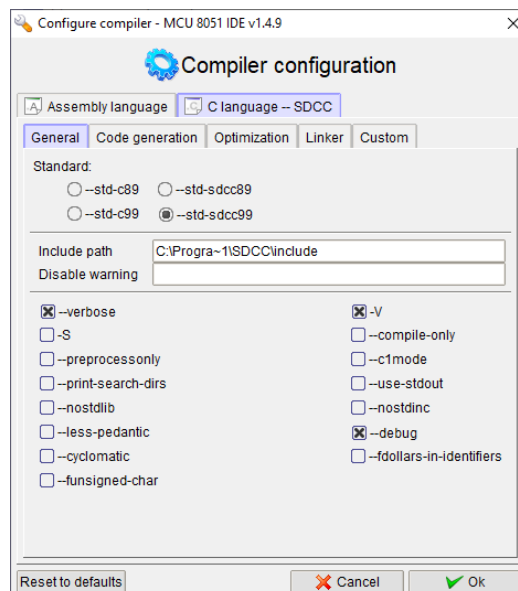
Zadanie 1

- a) W środowisku MCU 8051 IDE stwórz nowy projekt wprowadzając w oknie *New Project* ustawienia, jak na rysunku 1. W polu *Project directory* należy wpisać ścieżkę do folderu, w którym będą zapisywane pliki projektu. Można ją również wskazać z wykorzystaniem okna wyboru przez kliknięcie klawisza z ikoną .



Rys.1. Widok okna *New project* z zalecanymi ustawieniami w części *Processor*

- b) Z menu *Configure* wybierz *Configure Compiler*. W otwartym oknie, na zakładce *C language – SDCC* wybierz opcję `--std-sdcc99` oraz w polu *Include path* wpisz ścieżkę dostępu do folderu `include` znajdującego się w folderze instalacji kompilatora SDCC. Jeśli ścieżka do folderu instalacji kompilatora SDCC zawiera nazwy dłuższe niż 8 znaków, to może być konieczna jej specyfikacja jak na rysunku 2.



Rys.2. Ustawienia na zakładce *C language – SDCC*

- c) W oknie edycji kodu wpisz szablon kodu programu podany na listingu 1. Komentarze wskazują miejsca, w których powinny znaleźć się poszczególne elementy programu.

Listing 1. Szablon kodu programu

```
//=====
// Pliki nagłówkowe
#include <8052.h>

//=====
// Deklaracje zmiennych globalnych

//=====
// Prototypy funkcji

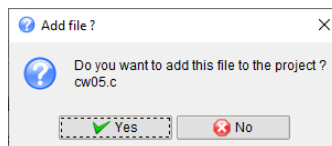
//=====
// Funkcje obsługi przerw

//=====
// Główna funkcja programu
void main (void)
{

    while (1);
}

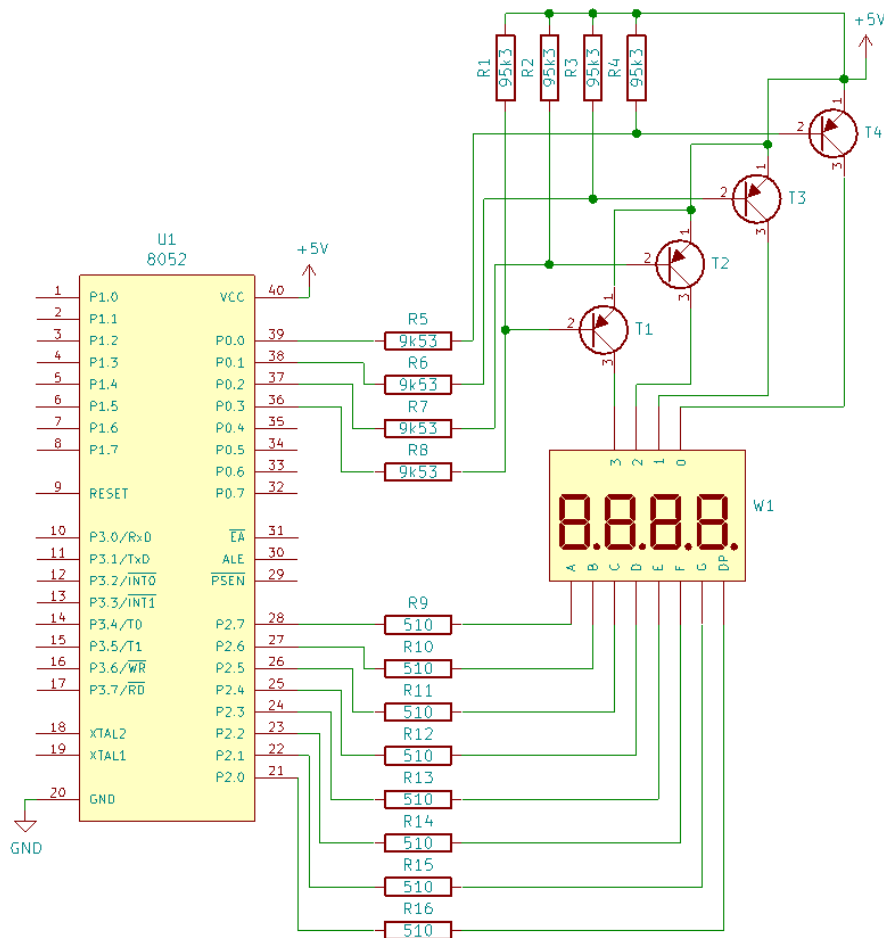
//=====
// Funkcje użytkownika
```

- d) Zapisz plik z kodem. W tym celu z menu *File* wybierz *Save as*. Wprowadzając nazwę pliku staraj się nie używać polskich znaków narodowych i spacji oraz ograniczyć się do 8 znaków. Plik zapisz z rozszerzeniem *.c. Po zatwierdzeniu zapisu pliku pojawi się oknie, jak na rysunku 3, w którym należy kliknąć klawisz *Yes*.





Rys.3. Widok okna z pytaniem, czy dołączyć zapisany plik do projektu

- e) Na podstawie schematu ideowego przedstawionego na rysunku 4, dokonaj podłączenia wirtualnego wyświetlacza LED do wyprowadzeń mikrokontrolera 8052. W tym celu wykorzystaj informacje przedstawione na slajdach z wykładu.
- f) W ustawieniach wyświetlacza LED ustaw interwał wygaszania wartość 200.



Rys.4. Schemat podłączenia wyświetlacza LED do mikrokontrolera



Zadanie 2

- Wykorzystując informacje przedstawione na wykładzie z przedmiotu zrealizuj wyświetlanie tej samej informacji na wszystkich modułach wyświetlacza LED w oparciu o nieskończoną pętlę programową.
- Dokonaj kompilacji programu przez kliknięcie klawisza z ikoną . W przypadku wystąpienia błędów składniowych usuń je i przeprowadź ponowną kompilację.
- Sprawdź poprawność działania programu uruchamiając go symulatorze wbudowanym w środowisko MCU8051 IDE. Nie rób tego w pracy krokowej, tylko przez kliknięcie klawisza . W przypadku nieprawidłowego działania programu zlokalizuj i usuń błędy, które takie działanie powodują.
- Uzupełnij komentarze w kodzie programu i spakuj wszystkie pliki projektu do archiwum ZIP o nazwie *cw6zad2.zip* w celu ich późniejszego przesłania.



Zadanie 3

- Na podstawie informacji przedstawionych na wykładzie zdefiniuj kody, które przesłane na port sterujący segmentami wyświetlaczy umożliwią prezentację cyfr od 0 do 9. Typ modułów LED (wspólna katoda lub anoda) wywniosuj ze schematu przedstawionego na rysunku 4.
- Zmodyfikuj kod programu z zadania 2 tak, aby możliwe było wyświetlanie różnych informacji na kolejnych modułach LED. W tym celu zadeklaruj 4-bajtową tablicę w wewnętrznej pamięci



danych (każdy bajt będzie przechowywał informację dla pojedynczego modułu LED) i wpisz do niej kody wybranych cyfr (muszą to być cztery różne cyfry) do wyświetlenia.

- c) Dokonaj kompilacji programu przez kliknięcie klawisza z ikoną . W przypadku wystąpienia błędów składniowych usuń je i przeprowadź ponowną kompilację.
- d) Sprawdź poprawność działania programu uruchamiając go symulatorze wbudowanym w środowisko MCU8051 IDE. Nie rób tego w pracy krokowej, tylko przez kliknięcie klawisza . W przypadku nieprawidłowego działania programu zlokalizuj i usuń błędy, które takie działanie powodują.
- e) Uzupełnij komentarze w kodzie programu i spakuj wszystkie pliki projektu do nowego archiwum ZIP o nazwie *cw6zad3.zip* w celu ich późniejszego przesłania.

Zadanie 4



- a) Zadeklaruj w pamięci kodu tablicę 10-elementową tablice i zainicjuj ją wartościami odpowiadającymi kodom cyfr 0..9. Dokonaj modyfikacji programu z poprzedniego zadania w taki sposób, aby do 4-bajtowej tablicy zapisywane były wartości cyfr do wyświetlenia (muszą to być cztery różne cyfry), natomiast dopiero podczas wyświetlania ich kody były pobierane ze zdefiniowanej tablicy kodów.
- b) Dokonaj kompilacji programu przez kliknięcie klawisza z ikoną . W przypadku wystąpienia błędów składniowych usuń je i przeprowadź ponowną kompilację.
- c) Sprawdź poprawność działania programu uruchamiając go symulatorze wbudowanym w środowisko MCU8051 IDE. Nie rób tego w pracy krokowej, tylko przez kliknięcie klawisza . W przypadku nieprawidłowego działania programu zlokalizuj i usuń błędy, które takie działanie powodują.
- d) Uzupełnij komentarze w kodzie programu i spakuj wszystkie pliki projektu do nowego archiwum ZIP o nazwie *cw6zad4.zip* w celu ich późniejszego przesłania.

Zadanie 5

- a) Dokonaj kolejnej modyfikacji programu tak, aby czas, co jaki odświeżany jest pojedynczy modułu LED wynosił 224 μ s. Wykorzystaj w tym celu przerwanie od jednostki czasowo-licznikowej T0 mikrokontrolera (każde wywołanie przerwania powinno odświeżać pojedynczy moduł LED).
- b) Dokonaj kompilacji programu przez kliknięcie klawisza z ikoną . W przypadku wystąpienia błędów składniowych usuń je i przeprowadź ponowną kompilację.
- c) Sprawdź poprawność działania programu uruchamiając go symulatorze wbudowanym w środowisko MCU8051 IDE. Nie rób tego w pracy krokowej, tylko przez kliknięcie klawisza . W przypadku nieprawidłowego działania programu zlokalizuj i usuń błędy, które takie działanie powodują.
- d) Uzupełnij komentarze w kodzie programu i spakuj wszystkie pliki projektu do nowego archiwum ZIP o nazwie *cw6zad5.zip* w celu ich późniejszego przesłania.

Zadanie 6

- a) Do tworzonego programu dodaj kod pozwalający na obsługę przerw generowanych przez jednostkę czasowo-licznikową T2 mikrokontrolera. Dobierz parametry pracy licznika T2 w taki sposób, aby możliwe było odliczanie czasu równego 1 ms.

- b) W ramach funkcji obsługi przerwania od licznika T2 zaimplementuj kod pozwalający na zliczanie 1-milisekundowych interwałów i prezentację na bieżąco tego wyniku na wyświetlaczu LED. W tym celu wykorzystaj 4-elementową tablicę z danymi do prezentacji cyfr na wyświetlaczu LED. Po zliczeniu 20 interwałów należy zliczanie realizować od 0.
- c) Dokonaj kompilacji programu przez kliknięcie klawisza z ikoną . W przypadku wystąpienia błędów składniowych usuń je i przeprowadź ponowną kompilację.
- d) Sprawdź poprawność działania programu uruchamiając go symulatorze wbudowanym w środowisko MCU8051 IDE. Nie rób tego w pracy krokowej, tylko przez kliknięcie klawisza . W przypadku nieprawidłowego działania programu zlokalizuj i usuń błędy, które takie działanie powodują.
- e) Uzupełnij komentarze w kodzie programu i spakuj wszystkie pliki projektu do nowego archiwum ZIP o nazwie *cw6zad6.zip* w celu ich późniejszego przesłania.

Sprawozdanie z ćwiczenia

Sprawozdanie z ćwiczenia powinno być dostarczone prowadzącemu zajęcia w określonej przez niego formie (pisemnej lub elektronicznej) i zawierać:

- opatrzony komentarzami kod źródłowy programu napisanego w ramach realizacji zadań 4 oraz 6,
- algorytmy wszystkich funkcji programu napisanego w ramach realizacji zadań 4 i 6,
- opis słowny wyjaśniający, w jaki sposób wyznaczono kody cyfr dla wyświetlacza LED,
- opis słowny uzasadniający przyjęte dla licznika T0 ustawienia (m.in. tryb pracy i wartości początkowe) w celu spełnienia założeń wymaganych w ramach zadania 5,
- opis słowny uzasadniający przyjęte dla licznika T2 ustawienia (m.in. tryb pracy i wartości początkowe) w celu spełnienia założeń wymaganych w ramach zadania 6.

Wraz ze sprawozdaniem należy prowadzącemu zajęcia wysłać również spakowane pliki programów z zadań od 2 do 6.

Literatura

- [1] Wykłady do przedmiotu.
- [2] Bogusz J.: „*Programowanie mikrokontrolerów 8051 w języku C w praktyce*”. Wydawnictwo BTC, Warszawa 2005.
- [3] Kernighan W., Ritchie D.M.: „*ANSI C*”. WNT, Warszawa 1992.
- [4] Rydzewski A.: „*Mikrokomputery jednocukładowe rodziny MCS-51*”. WNT, Warszawa 1992.
- [5] Starecki T.: „*Mikrokontrolery jednocukładowe rodziny 51*”. Wydawnictwo NOZOMI, Warszawa 1996.
- [6] Starecki T.: „*Mikrokontrolery 8051 w praktyce*”. Wydawnictwo BTC, Warszawa 2003.
- [7] Majewski J.: „*Programowanie mikrokontrolerów 8051 w języku C, pierwsze kroki*”. Wydawnictwo BTC, Warszawa 2005.