

POLITECHNIKA POZNAŃSKA

INSTYTUT AUTOMATYKI I INŻYNIERII INFORMATYCZNEJ

ZAKŁAD STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ



KOMUNIKACJA SZEREGOWA - UART

PROGRAMOWANIE MIKROPROCESORÓW

MATERIAŁY DO ZAJĘĆ LABORATORYJNYCH

DR INŻ. DOMINIK ŁUCZAK

DOMINIK.LUCZAK@PUT.POZNAN.PL

I. CEL

Celem zajęć jest zapoznanie się z działaniem komunikacji szeregowej oraz nabycie umiejętności programistycznych pozwalających na jej programową obsługę.

II. POLECENIA KOŃCOWE

Dokończ zadania nie zrealizowane w trakcie zajęć. Przygotuj raport z przeprowadzonych zajęć. Jeden raport może być przygotowany przez maksymalnie 2 osoby. W projekcie zamieść:

1. Opis rozwiązania problemu oraz jego implementację.
2. Fotografie ilustrujące poprawne działanie programu.
3. Opis działania własnych programów. Udokumentuj przeprowadzenie testów funkcjonalnych.
4. Ważne instrukcje programu należy zaopatrzyć stosownym komentarzem. Wszystkie przygotowane funkcje oraz zmienne globalne należy zaopatrzyć komentarzem zgodnym z składnią generatora dokumentacji (*Doxygen*).

Jako załącznik dołącz spakowane kody programów. Dla każdego zadania przygotuj osobny projekt. Raport należy przygotować i przekazać do kolejnego spotkania. Raporty przekazywane później nie będą oceniane.

Na ocenę raportu będą miały wpływ następujące elementy (łącznie 5 punktów):

1. spełnienie wymogów redakcyjnych (1p),
2. wykonanie, udokumentowanie oraz opis wykonanych zadań (2p),
3. zastosowanie prawidłowego warsztatu programistycznego (2p).

III. PRZYGOTOWANIE DO ZAJĘĆ

a) ZAPOZNANIE Z PRZEPISAMI BHP

Wszystkie informacje dotyczące instrukcji BHP laboratorium są zamieszczone w sali laboratoryjnej oraz u prowadzącego zajęcia. Wszystkie nieścisłości należy wyjaśnić z prowadzącym laboratorium. Wymagane jest zaznajomienie i zastosowanie do regulaminu.

Na zajęcia należy przyjść przygotowanym zgodnie z tematem zajęć. Obowiązuje również materiał ze wszystkich odbytych zajęć.

1. Obsługa UART Timer 1:
 - a. napisz szkielet programu z obsługą przerwania od UART,
 - b. wykorzystaj Timer 1 do określenia prędkości transmisji – określ przewodzący np. 9600,
 - c. ustaw liczbę bitów danych na 8 lub 9 bitów,
 - d. dodaj „echo” – dobierane znaki są wysyłane do nadawcy - sprawdź poprawność działania programu w debuggerze.
2. Obsługa UART Timer 2:

- a. wykorzystaj Timer 2 do określenia prędkości transmisji – określa prowadzący np. 2400,
 - b. ustaw liczbę bitów danych na 8,
 - c. sprawdź poprawność działania programu w debuggerze.
3. Obsługa UART Timer 1 oraz Timer 2:
- a. wykorzystaj Timer 1 oraz Timer 2 do określenia prędkości transmisji – określa prowadzący np. 9600 i 2400,
 - b. ustaw liczbę bitów danych na 8,
 - c. sprawdź poprawność działania programu w debuggerze.
4. Obsługa UART Timer 3:
- a. wykorzystaj Timer 3 do określenia prędkości transmisji – określa prowadzący np. 115200,
 - b. ustaw liczbę bitów danych na 8,
 - c. zaproponuj zestaw zapytań i odpowiedzi do sterowania diodami i uzyskaniem stanu wejść cyfrowych, np. *L1S* – ustawia pierwszą diodę, *L1R* - resetuje pierwszą diodę,
 - d. sprawdź poprawność działania programu w oknie debuggera.
 - e. Skorzystaj z emulatora null-modem do utworzenia pary portów szeregowych (COM4-COM5). Wykorzystując polecenie *assign*, dostępne w debuggerze Keila, przypisz port COM4 do UARTa. Otwórz terminal na porcie COM5. Sprawdź poprawność działania programu wykorzystując utworzone połączenie pomiędzy debuggerem a terminalem.
5. Prosta aplikacja wykorzystująca UART:
- a. rozszerz zestaw poleceń umożliwiając prostą parametryzację systemu np. 1) wprowadź okres czasu przełączania każdej z diod, 2) dodaj zgodę na pracę każdej diody, 3) ustaw domyślny stan diody w przypadku braku zgody na pracę przemienną.
 - b. sprawdź poprawność działania programu w debuggerze a następnie terminalu PC.