

Programowanie niskopoziomowe
Lista 2 - system przerwań

1. Napisz zręby
 - programu umożliwiającego bezpieczną obsługę przerwań
 - bezpiecznej procedury obsługi przerwaniaPrzedstaw stan stosu podczas obsługi przerwania
2. W programie co określony czas zwiększany jest licznik programowy, a jego wartość jest konwertowana na kod BCD w celu wyświetlenia. Jednocześnie odbierana jest transmisja danych z USART - dane pojawiają się w różnych odstępach czasu. W tablicy *czasys_transmisji* notowany jest stan licznika programowego w chwili transmisji. Załóżmy, że:
 - że *konwertuj_na_BCD_i_wyswietl* jest zdefiniowane i czasochłonne
 - *inicjalizuj_USART* oraz *obsługa_przerwan_USART_i_zapis_do_tablicy* są zdefiniowane.Twoim zadaniem jest więc stosunkowo proste: napisz obsługę licznika programowego (w tym wywołanie konwersji na BCD) z wykorzystaniem systemu przerwań timera, tak aby zminimalizować ryzyko utracenia danych z USART i przekłamań w tablicy *czasys_transmisji*.
3. Wyobraźmy sobie, że w powyższym przykładzie korzystamy z TIMER0, który ma wyższy priorytet w tabeli przerwań niż USART. Ponadto niech czas zliczany przez TIMER0 będzie porównywalny z czasem obsługi jego przerwania. W takiej sytuacji przerwanie z USART może nie zostać obsłużone przez bardzo długi czas. Aby rozwiązać ten problem zaproponuj rozwiązanie oparte o zagnieżdżanie przerwań.
4. W programie wychylenie joysticka zmienia procedurę obsługi przerwania od timera. Ponieważ nie można bezpośrednio modyfikować tablicy wektorów przerwań zaproponuj rozwiązanie z wykorzystaniem pamięci RAM.

Materiały:

ATMEL – (instrukcje procesora)	doc0856.pdf
(zegar systemowy, przerwania, timery)	doc7593.pdf
A. Pawluczuk – „Sztuka programowania mikrokontrolerów AVR”, rozdz. 12.1 (obsługa przerwań od upływu czasu)	

Na ćwiczeniach zaleca się przedstawienie kilku rozwiązań powyższych problemów.

Witold Dyrka
Marcin Masalski