



Instytut Teleinformatyki



Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska

Systemy Wbudowane

„FreeRTOS na płytce ewaluacyjnej AVR”

laboratorium: 03
autor: mgr inż. Dariusz Żelasko

Kraków, 2015

Spis treści

Spis treści	2
1. Wiadomości wstępne	3
2. Przebieg laboratorium	4
2.1. Zadanie 1. Na ocenę 3.0 (dst)	4
2.2. Zadanie 2. Na ocenę 4.0 (db)	4
2.3. Zadanie 3. Na ocenę 5.0 (bdb)	5

1. Wiadomości wstępne

FreeRTOS:

<http://www.freertos.org>

Nota katalogowa Atmega32:

www.atmel.com/images/doc2503.pdf

Dokumentacja zestawu ewaluacyjnego ZL15AVR

www.cyfronika.com.pl/uruchomieniowe/zl15avr.pdf

Wykorzystanie FSM w modelowaniu systemów sterowanych zdarzeniowo

elf2.pk.edu.pl

Implementacja FSM z wykorzystaniem Finie State Table

elf2.pk.edu.pl

Przykładowe aplikacje dla ZL15AVR

http://www.kamami.pl/dl/zl15avr_c_examples.zip

Uruchomienie projektu:

Wszelkiej modyfikacji kodu najlepiej dokonywać w pliku **FreeRTOS\kod\DEMO\main.c** przy użyciu dowolnego edytora tekstu. Do stworzenia pojedynczego wątku służy funkcja **xTaskCreate()**. Uruchomienie wszystkich zadeklarowanych wątków odbywa się przy użyciu funkcji **vTaskStartScheduler()**. Aby skompilować kod należy uruchomić linię komend, przejść do katalogu w którym znajduje się aplikacja i wydać polecenie **make**. Po kompilacji pojawi się plik **.hex** który można wgrać na płytkę przy użyciu kamprog. Aby usunąć niepotrzebne pliki przed kompilacją należy wydać polecenie **make clean**.

Literatura:

[1] Rob Williams „*Real-Time Systems Development*”, Elsevier, 2007

2. Przebieg laboratorium

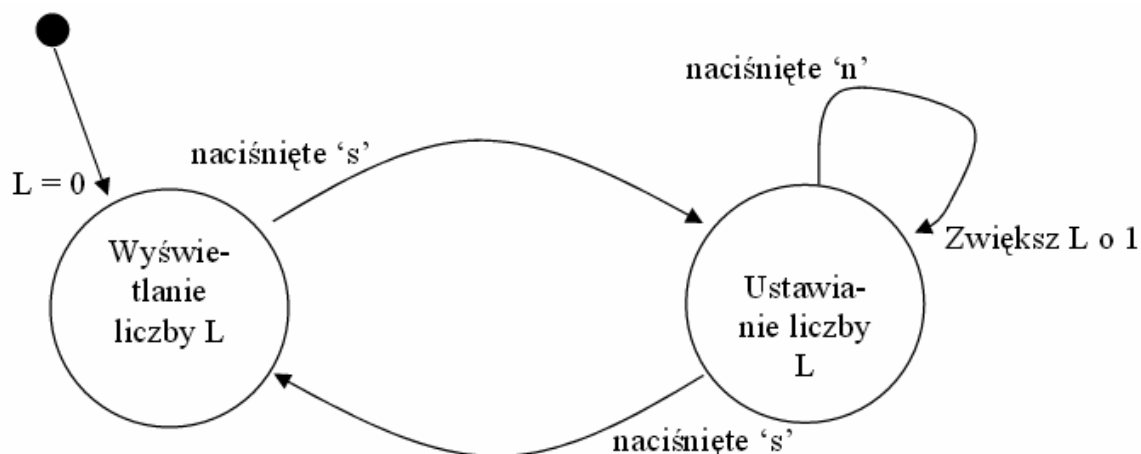
Druga część instrukcji zawiera zadania do praktycznej realizacji, które demonstrują zastosowanie technik z omawianego zagadnienia.

2.1. Zadanie 1. Na ocenę 3.0 (dst)

Proszę, odpowiednio modyfikując dostarczony kod, napisać program w którym będą działały dwa wątki - pierwszy będzie powodował zapalenie i gaszenie diod 0-3 co 5 ms. Drugi będzie powodował zapalenie i gaszenie diod 4-7 co 30 ms. Pamiętaj o dobraniu właściwych priorytetów dla poszczególnych wątków.

2.2. Zadanie 2. Na ocenę 4.0 (db)

Modyfikując otrzymany kod proszę zaimplementować FSM opisaną następującym diagramem:



W implementacji proszę użyć metody Finite State Table. Przejścia między stanami mają być generowane poprzez naciśnięcie przycisku SW0 (s) i SW1 (n). L należy wyświetlić binarnie na diodach LED0-LED7.

2.3. Zadanie 3. Na ocenę 5.0 (bdb)

Korzystając z metody FST i poniższego diagramu proszę zaimplementować zegar w którym godziny, minuty i sekundy są nastawiane. Aktualny czas (H i M) należy wyświetlić na wyświetlaczu 7-segmentowym – proszę użyć przykładowego kodu z pliku zl15avr_c_examples.zip, dostępnego na stronie producenta zestawu ewaluacyjnego. Przejścia między stanami mają być generowane poprzez naciśnięcie przycisku SW0 (s) i SW1 (n).

