



# Instytut Teleinformatyki



Wydział Fizyki, Matematyki i Informatyki  
Politechnika Krakowska

**Systemy Wbudowane**

---

**„FSM – implementacja na płytce ewaluacyjnej AVR”**

laboratorium: 02  
autor: dr hab. Zbysław Tabor, prof. PK

**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) .....	5
2.3. Zadanie 3. Na ocenę 5.0 (bdb) .....	6

# 1. Wiadomości wstępne

Przerwania ATmega32:

[https://sites.google.com/site/zbislawtabor/dydaktyka/systemy-wbudowane/przerwania\\_avr](https://sites.google.com/site/zbislawtabor/dydaktyka/systemy-wbudowane/przerwania_avr)

Nota katalogowa Atmega32:

[www.atmel.com/images/doc2503.pdf](http://www.atmel.com/images/doc2503.pdf)

Dokumentacja zestawu ewaluacyjnego ZL15AVR

[www.cyfronika.com.pl/uruchomieniowe/zl15avr.pdf](http://www.cyfronika.com.pl/uruchomieniowe/zl15avr.pdf)

Wykorzystanie FSM w modelowaniu systemów sterowanych zdarzeniowo

[elf2.pk.edu.pl](http://elf2.pk.edu.pl)

Implementacja FSM z wykorzystaniem Finie State Table

[elf2.pk.edu.pl](http://elf2.pk.edu.pl)

## Literatura:

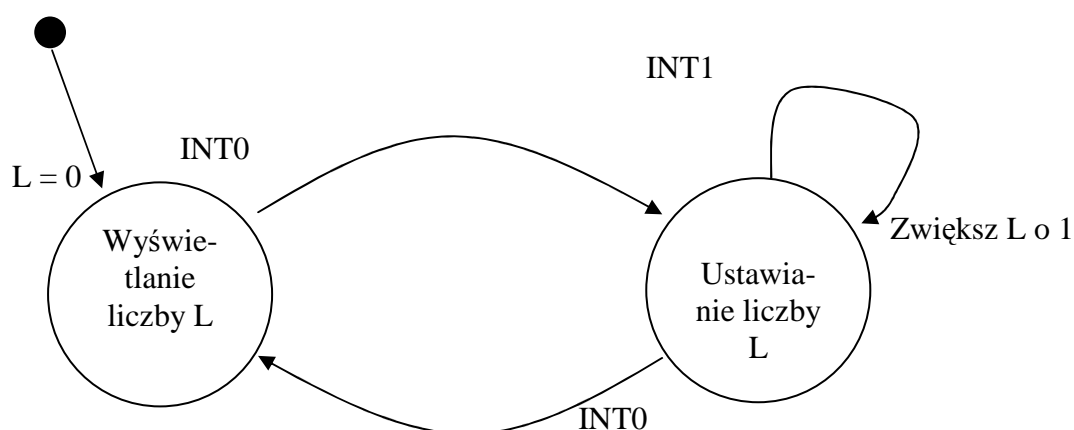
[1] Rob Williams „*Real-Time Systems Development*”, Elsevier, 2007

## 2. Przebieg laboratorium

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

### 2.1. Zadanie 1. Na ocenę 3.0 (dst)

W oparciu o kod z Listingu 1 proszę zaimplementować FSM opisaną następującym diagramem:



W implementacji proszę użyć metody Finite State Table. Przejścia między stanami mają być generowane przerwaniem zewnętrznym INT0 i INT1. W funkcji main w pętli nieskończonej mają być realizowane funkcjonalności związane z dekodowaniem zdarzenia oraz wykonaniem akcji skojarzonej z przejściem między stanami. Wartość L należy wyświetlać na diodach.

#### Listing 1 Obsługa przerwania zewnętrznego

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

int main(void)
```

```
{
    unsigned char licz;

    DDRD &= ~(1<<PD2);
    PORTD=0xFF;
    GICR = 1<<INT0;           // Enable INT0
    MCUCR = 1<<ISC01 | 1<<ISC00; // Trigger INT0 on rising edge

    sei();                    //Enable Global Interrupt

    DDRB = 0xFF;
    PORTB=0;
    licz=0;

    while(1)
    {
        _delay_ms(1000);
        PORTB = (1<<licz);
        licz=++licz%8;
    }
}

//Interrupt Service Routine for INT0
ISR(INT0_vect)
{
    unsigned char i, temp;

    // _delay_ms(500); // Software debouncing control

    temp = PORTB; // Save current value on DataPort

    // do some stuff

    PORTB = temp; //Restore old value to DataPort
}
```

## 2.2. Zadanie 2. Na ocenę 4.0 (db)

Proszę zastąpić liczbę L z zadania 1 licznikiem sekund. Licznik ma być powiększany co 1 sekundę (niezależnie od stanu, w którym znajduje się FSM). Do generowania zdarzenia od timera należy użyć przerwania od timera. Wartość L należy wyświetlić na diodach.

### 2.3. Zadanie 3. Na ocenę 5.0 (bdb)

Zegar: korzystając z metody FST i poniższego diagramu proszę zaimplementować zegar w którym minuty i sekundy są nastawiane. Aktualny czas należy wyświetlić na wyświetlaczu 7-segmentowym – proszę użyć przykładowego kodu z pliku zl15avr\_c\_examples.zip, dostępnego na stronie producenta zestawu ewaluacyjnego.

