



Instytut Teleinformatyki



Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska

Mikroprocesory i mikrokontrolery

„System przerwań procesora AVR”

laboratorium: 09
autor: mgr inż. Katarzyna Smelcerz

Kraków, 2015

Spis treści

Spis treści	2
1. Wiadomości wstępne	3
1.1. Niezbędne wiadomości	3
1.2. Założenie nowego projektu w środowisku AVR Studio4	Błąd! Nie zdefiniowano zakładki.
1.3. Konfiguracja portów	Błąd! Nie zdefiniowano zakładki.
2. Przebieg laboratorium	5
2.1. Zadanie 1. Na ocenę 3.0 (dst)	5
2.2. Zadanie 2. Na ocenę 4.0 (db)	9
2.3. Zadanie 3. Na ocenę 5.0 (bdb)	9

1. Wiadomości wstępne

Pierwsza część niniejszej instrukcji zawiera podstawowe wiadomości teoretyczne dotyczące omawianego tematu. Poznanie tych wiadomości umożliwi prawidłowe zrealizowanie praktycznej części laboratorium.

1.1. Niezbędne wiadomości

Pamięć programu, wewnętrzna pamięć RAM, mapa pamięci, rejestry, tryby adresowania oraz lista instrukcji, dostępna w dokumentacji:
<http://www.atmel.com/images/doc2503.pdf>

Schemat płytki ewaluacyjnej AVRZL15
<http://www.cyfronika.com.pl/uruchomieniowe/zl15avr.pdf>

Bieżące laboratorium jest przeprowadzane w środowisku AVR Studio4. Do fizycznego zaprogramowania mikrokontrolera używany jest programator (np. KamPROG)

1.2. Timer0 Atmega32

Główne funkcje Timera0:

- jednostka porównująca
- Timer z funkcją Compare Match (Auto Reload)
- PWM
- generator częstotliwości
- licznik zdarzeń zewnętrznych
- 10-bit preskaler zegara
- tryb Overflow (TOV0) i Compare Match (OCF0) dla przerwania od Timera

Ustawienia preskalera dla Timera0 wpisywane do rejestru Stand_TCCR0

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{I/O} /(No prescaling)
0	1	0	clk _{I/O} /8 (From prescaler)
0	1	1	clk _{I/O} /64 (From prescaler)
1	0	0	clk _{I/O} /256 (From prescaler)
1	0	1	clk _{I/O} /1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Dodatkowe, bardziej szczegółowe informacje dotyczące Timerów znajdują się w dokumentacji podanej w rozdziale 1.1.

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)

Należy przeanalizować pracę mikrokontrolera poprzez wykonanie symulacji krokowej kodu przedstawionego na **Listing 1**, a następnie uruchomić program na płycie ZL15AVR. Proszę zwrócić uwagę, na to, kiedy ustawiają się flagi przerwania. Proszę skomentować program, tak aby wszystkie instrukcje stały się jasne.

Listing 1

```
.nolist
.include "m32def.inc"
.list
.listmac

.macro    makro1
    push  acc
    in    acc,sreg
    push  acc
.endm
;-----
.macro    makro2
    pop   acc
    out  sreg,acc
    pop  acc
.endm
;-----
.def     acc    =    r16
.def     acc2   =    r17
;-----
.equ     Joystick    =    PORTA
.equ     Diody       =    PORTB
.equ     JoystickPin =    0
```

```

.equ   DiodyPin      =      0
;-----
.equ   JoystickPDirection =      Joystick - 1
.equ   DiodyPDirection =      Diody - 1
.equ   JoystickPininput=      Joystick - 2
;-----

.equ   TimerData     =      0xA0
.equ   DivideConst   =      150           //
.equ   LEDTimeInterv =      40           //
;-----

;-----
.equ           Stand_TIMSK=      0b00000001

;           Timer/counter Interrupt Mask           ||| |
;           || | +---- TOIE0 : Timer/Counter 0 Overflow Interrupt
Enable
;           +---- OCIE0: Timer/Counter0 capture
compare enable
;           || +----- TICIE1 : Timer/Counter1 Input Capture
Interrupt Enable
;           ||+----- OCIE1B : Timer/counter1 Output Compare
B Match Interrupt Enable
;           |+----- OCE1A : Timer/counter1 Output Compare
A Match Interrupt Enable
;           +----- TOIE1 : Timer/Counter1 Overflow Inter-
rupt Enable
;-----

.equ           Stand_TCCR0=      0b00000011
;           Timer/Counter0 Control Register           |||
;           ||+-- CS00
;           |+--- CS01
;           +---- CS02
;
;           CS02  CS01  CS00
;           0    0    0    timer/counter0 is stopped
;           0    0    1    clk
;           0    1    0    clk/8
;           0    1    1    clk/64
;           1    0    0    clk/256
;           1    0    1    clk/1024
;           1    1    0    external pin T0; falling edge
;           1    1    1    external pin T0; rising edge
;-----

.dseg

```

```

;*****
;
TimeDivider : .byte 1 ;
LEDTimer : .byte 1 ;
;-----

.cseg
;*****
;*****
.org 0
rjmp ResetProcessor
;-----

.org OVF0addr ;Overflow0 Interrupt Vector Address
rjmp OVF0Interrupt
;-----

OVF0Interrupt : ;PROCEDURE OVF0Interrupt ;
;***** ;

makro1 ;
push acc ;

;
;-----

lds acc, TimeDivider
or acc, acc
breq xx1
dec acc
sts TimeDivider, acc

;
;-----

sbi Diody,DiodyPin ;
rjmp koniec ;

;
;-----

xx1: cbi Diody,DiodyPin ;

koniec:pop acc
makro2 ;

```

```

        reti                                ;END ;

;reti to rozkaz interrupt return, odtwarza licznik rozkazów
;-----

;-----
InitEnvir      :                                ;PROCEDURE InitEnvir ;
;*****
;BEGIN
        cbi    JoystickPDirection,JoystickPin    ;
        sbi    Joystick,JoystickPin              ;
        sbi    DiodyPDirection,DiodyPin         ;
        cbi    Diody,DiodyPin                   ;
        ldi    acc,Stand_TIMSK                  ;
        out    TIMSK,acc                         ;
        ldi    acc,Stand_TCCR0                  ;
        out    TCCR0,acc                        ;
        ldi    acc,TimerData                    ; ;
        out    TCNT0,acc                        ;
        ret                                       ; ;
;-----

InitVariable   :                                ;PROCEDURE InitVariable ;
;*****
;
        ldi    acc,DivideConst                  ;
        sts    TimeDivider,acc                  ;
        ldi    acc,0                            ;
        sts    LEDTimer,acc                     ;
        ret                                       ;
;-----

Zapal_diode   :                                ;PROCEDURE Zapal_diode ;
;*****
;
        ldi    acc,LEDTimeInterv               ;
        sts    TimeDivider, acc                 ;
        ret                                       ;END ;
;-----

Czekaj_na_joystick :                                ;PROCEDURE Czekaj_na_joystick ;
;*****
;BEGIN
Wcisniety     :                                ;
;BEGIN
        in     acc,JoystickPininput            ;
        bst    acc,JoystickPin                 ;
        brts   Wcisniety                       ; ;
        ret                                       ;
;-----

Czekaj_na_zwolnienie_joysticka :                                ;PROCEDURE Cze-
kaj_na_zwolnienie_joysticka ;
;*****
;BEGIN
Zwolniony     :                                ;
;BEGIN

```



```

        in     acc,JoystickPinput      ;
        bst   acc,JoystickPin         ;
        brtc  Zwolniony               ;
        ret                               ;
;-----
ResetProcessor    :                      ;BEGIN
        ldi   acc,255                  ;LOW(RAMEND)      ;
        out   SPL,acc                  ;
        rcall InitEnvir                ;
        rcall InitVariable              ;
        sei                               ;;
Main_0            :                      ;
        rcall Czekaj_na_joystick       ;
        rcall Zapal_diode              ;
        rcall Czekaj_na_zwolnienie_joysticka ;

        rjmp  Main_0                  ;
                               ;END.
;-----
.exit

```

2.2. Zadanie 2. Na ocenę 4.0 (db)

Proszę zapoznać się z programem z Listingu 1 oraz zmodyfikować go, w taki sposób, aby dioda gasiła się z innym opóźnieniem. Dodatkowo proszę usunąć definicję zmiennych `acc` oraz `acc2` oraz makra (`.macro`) – wprowadzić odpowiednie zmiany do programu, tak aby się kompilował. Jakie różnice można zauważyć między architekturą RISK i CISK? Podaj przykłady takich rdzeni.

2.3. Zadanie 3. Na ocenę 5.0 (bdb)

Proszę program z Listingu 1 zmodyfikować w taki sposób, aby Timer0 działał w trybie Compare Match. Proszę również zmienić czas, po którym dioda gaśnie. Jakie różnice można zauważyć między architekturą RISK i CISK? Podaj przykłady takich rdzeni.