



Zachodniopomorski  
Uniwersytet  
Technologiczny

**WYDZIAŁ ELEKTRYCZNY**  
Katedra Inżynierii Systemów, Sygnałów i Elektroniki

**LABORATORIUM**

**TECHNIKA MIKROPROCESOROWA**

**SYSTEM PRZERWAŃ  
ATmega 32**

Opracował:

mgr inż. Andrzej Biedka

W mikrokontrolerze Atmega 32 zastosowano wektoryzowany system przerwań. Tablica 3-1 zawiera listę wektorów.

Tablica 3-1. Zestawienie wektorów przerwań uC Atmega 32

Lp	Źródło przerwania		Adres obsługi	Priorytet
1	RESET	Pin RESET, POR, Brown-out Reset, Watchdog Reset	0	najwyższy
2	INT0	zewnętrzne nr 0	0x002	
3	INT1	zewnętrzne nr 1	0x004	
4	INT2	zewnętrzne nr 2	0x006	
5	TIMER2COMP	zrównanie stanu rejestrów TCNT2 i OCR2	0x008	
6	TIMER2OVF	przepełnienie timera 2	0x00A	
7	TIMER1CAPT	zatrzaśnięcie zawartości timera 1	0x00C	
8	TIMER1COMPA	zrównanie stanu rejestrów TCNT1 i OCR1A	0x00E	
9	TIMER1COMPB	zrównanie stanu rejestrów TCNT1 i OCR1B	0x010	
10	TIMER1OVF	przepełnienie timera 1	0x012	
11	TIMER0COMP	zrównanie stanu rejestrów TCNT0 i OCR0	0x014	
12	TIMER0OVF	przepełnienie timera 0	0x016	
13	SPI, STC	Zakończenie transmisji SPI	0x018	
14	USART, RXC	Zakończenie odbioru znaku USART	0x01A	
15	USART UDRE	Zwolnienie bufora nadawczego	0x01C	
16	USART, TXC	Zakończenie nadawania znaku USART	0x01E	
17	ADC	Zakończenie przetwarzania A/C	0x020	
18	EE_RDY	Pamięć EEPROM gotowa	0x022	
19	ANA_COMP	Zmiana stanu komparatora	0x024	
20	TWI	Interfejs TWI	0x026	
21	SPM_RDY	Pamięć programu gotowa	0x028	najniższy

Zezwolenie na przyjęcie przerwania wymaga ustawienia bitu zezwolenia globalnego **I** umieszczonego w rejestrze statusu **SREG** oraz bitów zezwoleń właściwych dla każdego ze źródeł.

### SREG

MSB						LSB	
I	T	H	S	V	N	Z	C

**I** - bit globalnego zezwolenia na przerwania

W kompilatorze GCC zamiast ustawiania bitu I w rejestrze SREG można wywołać makro **sei()**;

Bit y zezwoleń na przerwania od timerów umieszczone są w rejestrze **TIMSK**:

MSB						LSB	
<b>OCIE2</b>	<b>TOIE2</b>	<b>TICIE1</b>	<b>OCIE1A</b>	<b>OCIE1B</b>	<b>TOIE1</b>	<b>OCIE0</b>	<b>TOIE0</b>

- OCIE2** - zrównanie stanu rejestrów TCNT2 i OCR2
- TOIE2** - przepełnienie timera 2
- TICIE1** - zatrzaśnięcie zawartości timera 1
- OCIE1A** - zrównanie stanu rejestrów TCNT1 i OCR1A
- OCIE1B** - zrównanie stanu rejestrów TCNT1 i OCR1B
- TOIE1** - przepełnienie timera 1
- OCIE0** - zrównanie stanu rejestrów TCNT2 i OCR0
- TOIE0** - przepełnienie timera 0

Bit y zezwoleń na przerwania od sygnałów zewnętrznych umieszczone są w rejestrze **GICR**:

MSB						LSB	
<b>INT1</b>	<b>INT0</b>	<b>INT2</b>	-	-	-	<b>IVSEL</b>	<b>IVCE</b>

- INT1** - zezwolenie na przerwanie od **INT1**
- INT0** - zezwolenie na przerwanie od **INT0**
- INT2** - zezwolenie na przerwanie od **INT2**

Flagi zgłoszenia przerwania znajdują się w rejestrze **GIFR**:

MSB						LSB	
<b>INTF1</b>	<b>INTF0</b>	<b>INTF2</b>	-	-	-	-	-

- INTF1** - zgłoszenie przerwania od **INT1**
- INTF0** - zgłoszenie przerwania od **INT0**
- INTF2** - zgłoszenie przerwania od **INT2**

Flagi zgłoszenia przerwania są kasowane automatycznie w funkcji obsługi przerwania. Można też flagę skasować wpisując stan '1' na jej pozycję. W trybie zgłoszenia przerwania poziomem flaga jest zawsze kasowana.

Wejścia INT0, INT1 mogą wywoływać przerwania zależnie od zmian stanu sygnału.

Bity sterujące sposobem reakcji na zmianę sygnału umieszczone są w rejestrze **MCUCR**:

MSB				LSB			
SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00

ISC11 ISC01	ISC10 ISC00	Opis
0	0	Niski poziom sygnału <b>INTx</b> wywołuje żądanie obsługi przerwania
0	1	Zmiana stanu na wejściu <b>INTx</b> wywołuje żądanie obsługi przerwania
1	0	Opadające zbocze sygnału <b>INTx</b> wywołuje żądanie obsługi przerwania
1	1	Narastające zbocze sygnału <b>INTx</b> wywołuje żądanie obsługi przerwania

/\*

Program generatora impulsów o częstotliwości 3Hz, z wykorzystaniem przerw

\*/

```
#include <avr/io.h>
```

```
#include <avr/interrupt.h>
```

```
#define PortLED PORTB
```

```
#define Czas166ms 65536 - 2604 // okres przebiegu = 333,333ms, 1/2 okresu = 166,667ms
```

```
int main(void)
```

```
{
```

```
DDRB = 0xFF; // PORTB ustaw na wyjście
```

```
TCCR1B = (1 << CS12) | (1 << CS10); // preskaler = 1024, tryb Normal
```

```
TCNT1 = Czas166ms;
```

```
TIMSK = (1 << TOIE1); // odblokowanie przerwania od przepełnienia timera 1
```

```
sei(); // globalne odblokowanie przerw
```

```
while(1);
```

```
}
```

```
ISR(TIMER1_OVF_vect)
```

```
{
```

```
TCNT1 = Czas166ms; // odtworzenie czasu timera
```

```
PortLED ^= 0xF; // neguj wybrane bity
```

```
}
```

#### Zadania:

1. Zmodyfikować powyższy program wykorzystując timer 0 pracujący w trybie CTC.
2. Napisać program generatora o założonych parametrach wykorzystujący dwa timery.
3. Napisać program sterowania jasnością czterech sąsiednich diod LED metodą PWM, wykorzystując timer pracujący w trybie Fast PWM oraz przerwanie od sygnału zewnętrznego. Ustalić cztery poziomy jasności zmieniane cyklicznie co 0,50 sekundy. Dla oceny zmian jasności cztery pozostałe diody powinny być stale załączone.

Tablica 3-2. Zestawienie nazw wybranych wektorów przerwań uC Atmega 32 stosowanych w kompilatorze AVR-GCC

Lp	Źródło przerwania		Nazwa AVR-GCC
1	INT0	zewnętrzne nr 0	INT0_vect
2	INT1	zewnętrzne nr 1	INT1_vect
3	INT2	zewnętrzne nr 2	INT2_vect
4	TIMER2COMP	zrównanie stanu rejestrów TCNT2 i OCR2	TIMER2_COMP_vect
5	TIMER2OVF	przepełnienie timera 2	TIMER2_OVF_vect
6	TIMER1CAPT	zatrzaśnięcie zawartości timera 1	TIMER1_CAPT_vect
7	TIMER1COMPA	zrównanie stanu rejestrów TCNT1 i OCR1A	TIMER1_COMPA_vect
8	TIMER1COMPB	zrównanie stanu rejestrów TCNT1 i OCR1B	TIMER1_COMPB_vect
9	TIMER1OVF	przepełnienie timera 1	TIMER1_OVF_vect
10	TIMER0COMP	zrównanie stanu rejestrów TCNT0 i OCR0	TIMER0_COMP_vect
11	TIMER0OVF	przepełnienie timera 0	TIMER0_OVF_vect
12	USART, RXC	Zakończenie odbioru znaku USART	USART_RXC_vect
13	USART UDRE	Zwolnienie bufora nadawczego	USART_UDRE_vect
14	USART, TXC	Zakończenie nadawania znaku USART	USART_TXC_vect
15	ADC	Zakończenie przetwarzania A/C	ADC_vect
16	ANA_COMP	Zmiana stanu komparatora	ANA_COMP_vect