



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΪΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΑΘΗΝΑ 25 Απριλίου 2024

**2η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**  
**ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"**

Εξέταση – Επίδειξη: Παρασκευή 17/5/2024.

**Εισαγωγή**

Μια χρήσιμη δυνατότητα των Μικροελεγκτών είναι η άμεση ανταπόκρισή τους σε εξωτερικές συνθήκες. Η ανταπόκριση αυτή επιτυγχάνεται με την εκμετάλλευση του συστήματος διακοπών τους. Κάθε μικροελεγκτής είναι εφοδιασμένος με μια ή περισσότερες εισόδους διακοπών. Η ενεργοποίηση μιας εισόδου διακοπής υποχρεώνει το μικροελεγκτή να σταματά άμεσα την τρέχουσα εργασία και να εκτελεί τον κώδικα που υπάρχει σε μια προκαθορισμένη διεύθυνση, που ονομάζεται διάνυσμα διακοπής. Στο σημείο αυτό τοποθετείται συνήθως μια ρουτίνα εξυπηρέτησης διακοπής (διαφορετική για κάθε εφαρμογή). Μετά το τέλος της ρουτίνας εξυπηρέτησης διακοπής, ο Μικροελεγκτής συνεχίζει την εργασία που διέκοψε, επιστρέφοντας στο σημείο ακριβώς που είχε διακοπεί.

**Καταχωρητές ρύθμισης των εξωτερικών διακοπών στον ATmega328PB**

Για τη λειτουργία του συστήματος διακοπών κάθε Μικροελεγκτή είναι απαραίτητη αρχικά η ενεργοποίηση σημαίων και επιλογών, που καθορίζουν τον ακριβή τρόπο λειτουργίας.

Στον Μικροελεγκτή AVR ATmega328PB, η επιλογή του επιπέδου ενεργοποίησης των εξωτερικών διακοπών γίνεται δια μέσω του καταχωρητή EICRA (offset 0x69), γράφοντας κατάλληλες τιμές στα τέσσερα λιγότερα σημαντικά ψηφία, σύμφωνα με τους παρακάτω πίνακες:

EICRA: 

				ISC11	ISC10	ISC01	ISC00
--	--	--	--	-------	-------	-------	-------

ISC11	ISC10	Περιγραφή
0	0	Διακοπή στη χαμηλή στάθμη του INT1
0	1	Διακοπή σε κάθε αλλαγή στάθμης του INT1
1	0	Διακοπή στην κατερχόμενη ακμή του INT1
1	1	Διακοπή στην ανερχόμενη ακμή του INT1

ISC01	ISC00	Περιγραφή
0	0	Διακοπή στη χαμηλή στάθμη του INT0
0	1	Διακοπή σε κάθε αλλαγή στάθμης του INT0
1	0	Διακοπή στην κατερχόμενη ακμή του INT0
1	1	Διακοπή στην ανερχόμενη ακμή του INT0

Η επίτρεψη των εξωτερικών διακοπών ενεργοποιείται γράφοντας στον καταχωρητή EIMSK (offset 0x3D) την τιμή 1 στο ψηφίο που αντιστοιχεί στην είσοδο διακοπής που επιθυμούμε να επιτρέψουμε, σύμφωνα με το παρακάτω σχήμα:

<b>EIMSK:</b>						INT1	INT0
---------------	--	--	--	--	--	------	------

Ο καταχωρητής EIFR(offset 0x3C) περιέχει τις σημαίες των διακοπών INT0 και INT1. Κάθε μια από αυτές τίθεται όταν συμβεί η αντίστοιχη διακοπή και μηδενίζεται μετά την εξυπηρέτηση της διακοπής. Εναλλακτικά, οποιαδήποτε από αυτές τις σημαίες μπορεί να μηδενιστεί γράφοντας την τιμή 1 σε αυτήν.

<b>EIFR:</b>						INTF1	INTF0
--------------	--	--	--	--	--	-------	-------

Επίσης, απαραίτητη είναι και η εκτέλεση της εντολής SEI, που θέτει την καθολική σημαία επίτρεψης των διακοπών (I) στον SREG (Status Register). Η εντολή που ακολουθεί την SEI θα εκτελεστεί πριν από την εκτέλεση τυχόν εκκρεμών διακοπών.

#### Είσοδοι εξωτερικών διακοπών

Στον Μικροελεγκτή AVR ATmega328P, ο ακροδέκτης PD2, εκτός από ακροδέκτης εισόδου/εξόδου γενικής χρήσης, λειτουργεί εναλλακτικά και ως εξωτερική είσοδος της διακοπής INT0.

Ομοίως ο ακροδέκτης PD3 λειτουργεί εναλλακτικά και ως εξωτερική είσοδος της διακοπής INT1.

#### Διανύσματα εξωτερικών διακοπών

Το διάνυσμα της διακοπής INT0 είναι 0x002.

Το διάνυσμα της διακοπής INT1 είναι 0x004.

#### Αποθήκευση / Επαναφορά του SREG στη στοίβα

Ο καταχωρητής κατάστασης (SREG) δεν αποθηκεύεται αυτόματα κατά την είσοδο σε μια ρουτίνα διακοπής, ούτε επαναφέρετε όταν επιστρέφει η ρουτίνα διακοπής. Αυτό πρέπει να αντιμετωπιστεί από λογισμικό:

##### A) Αποθήκευση του SREG

```
push r0;           // Push the register r0 to make space for SREG
in r0, SREG;       // Read the value of SREG into a register r0
push r0;           // Push the value of SREG (r0) onto the stack
```

##### B) Επαναφορά του SREG

```
pop r0;            // Pop the value of SREG (r0) from the stack into the register r0
out SREG, r0;      // Write the value of the register r0 back to SREG
```

Οποιοσδήποτε από τους 32 καταχωρητές γενικής χρήσης (r0 – r31) μπορεί να χρησιμοποιηθεί στη θέση του r0.

### Παράδειγμα 2.1 Assembly κώδικας χειρισμού εξωτερικών διακοπών

Ο κώδικας παρακάτω ενεργοποιεί τις διακοπές στην είσοδο INT0 κατά την ανερχόμενη ακμή.

Στη ρουτίνα εξυπηρέτησης της εξωτερικής διακοπής INT0, ανάβουν τα led της θύρας PORTB για 500mS.

Πρέπει να δοθεί προσοχή στο γεγονός ότι ο I/O καταχωρητής EICRA έχει η διεύθυνση μεγαλύτερη από 0x60, οπότε χρησιμοποιείται η εντολή STS και όχι η εντολή OUT.

```
.include "m328PBdef.inc"
.def number=r16

.org 0x0
    rjmp reset

.org 0x2
    rjmp isr0

reset:
    ldi r24, LOW(RAMEND)
    out SPL, r24
    ldi r24, HIGH(RAMEND)
    out SPH, r24

    ser r24
    out DDRB,r24          ; Init PORTB as output

    clr r24
    out DDRD,r24          ; Init PORTD as input

;Interrupt on rising edge of INT0 pin
    ldi r24, (1<<ISC01) | (1<<ISC00)
    sts EICRA, r24

;Enable the INT0 interrupt
    ldi r24, (1<<INT0)
    out EIMSK, r24

    sei                  ; Enable general flag of interrupts

    clr r24
    out PORTB, r24       ; Clear all pins of PORTB

main0:
    rjmp main0           ; main0 loop code do nothing

;External interrupt 0 service routine
isr0:
    push r23
    push r24
    push r25
```

```

in r25, SREG
push r25                ; save r23, r24, 25, SREG to stack

ldi r23, 0xFF           ; Set all pins of PORTB
out PORTB, r23

;Delay 500 mS
ldi r24, low(16*500)    ; Init r25, r24 for delay 500 mS
ldi r25, high(16*500)   ; CPU frequency = 16 MHz
delay1:
ldi r23, 249            ; (1 cycle)
delay2:
dec r23                 ; 1 cycle
nop                     ; 1 cycle
brne delay2             ; 1 or 2 cycles
sbiw r24, 1             ; 2 cycles
brne delay1             ; 1 or 2 cycles

clr r23
out PORTB, r23          ; Clear all pins of PORTB

ldi r24, (1 << INTF0)
out EIFR, r24           ; Clear external interrupt 0 flag

pop r25
out SREG, r25
pop r25
pop r24
pop r23                 ; Retrieve r23, r24, 25, SREG from stack

reti                    ; Return from interrupt

```

## Παράδειγμα 2.2 Ένα πρόγραμμα με εξωτερικά Interrupts σε γλώσσα C

Το πρόγραμμα που φαίνεται στην επόμενη σελίδα αποτελείται από τον κώδικα της κυρίας ρουτίνας, ο οποίος αναβοσβήνει συνεχώς τα led της PORTB, με καθυστέρηση 500mS. Επίσης ενεργοποιεί τις εξωτερικές διακοπές INT0 και INT1 κατά την ανερχόμενη ακμή τάσης στα αντίστοιχα pins.

Η ρουτίνα εξυπηρέτησης της διακοπής INT0 ανάβει τα led της PORTC για 2 Sec και μηδενίζει τη σημαία της διακοπής INT0.

Η ρουτίνα εξυπηρέτησης της διακοπής INT1 ανάβει τα led της PORTC για 3 Sec και μηδενίζει τη σημαία της διακοπής INT1.

```

#define F_CPU 16000000UL
#include<avr/io.h>
#include<avr/interrupt.h>
#include<util/delay.h>

ISR(INT0_vect)      // External INT0 ISR
{
    PORTC=0xFF;      // Turn on all LEDs of PORTC
    _delay_ms(2000); // for 2000ms
    PORTC=0x00;
    EIFR = (1 << INTF0); // Clear the flag of interrupt INTF0
}

ISR(INT1_vect)      // External INT1 ISR
{
    PORTC=0xFF;      // Turn on all LEDs of PORTC
    _delay_ms(3000); // for 3000ms
    PORTC=0x00;
    EIFR = (1 << INTF1); // Clear the flag of interrupt INTF1
}

int main(){
    // Interrupt on rising edge of INT0 and INT1 pin
    EICRA=(1<<ISC11) | (1<<ISC10) | (1<<ISC01) | (1<<ISC00);

    // Enable the INT0 interrupt (PD2), INT1 interrupt (PD3))
    EIMSK=(1<<INT0) | (1<<INT1);

    sei();           // Enable global interrupts

    DDRB=0xFF;       // Set PORTB as output
    DDRC=0xFF;       // Set PORTC as output

    PORTC=0x00;      // Turn off all LEDs of PORTC

    while(1)
    {
        PORTB=0x00;  // Turn off all LEDs of PORTC
        _delay_ms(500); // Delay 500 mS
        PORTB=0xFF;  // Turn on all LEDs of PORTC
        _delay_ms(500); // Delay 500 mS
    }
}

```

## Τα ζητούμενα της 2<sup>ης</sup> εργαστηριακής άσκησης

### Ζήτημα 2.1

Να υλοποιηθεί κώδικας Assembly για το μικροελεγκτή ATmega328PB, που απαριθμεί το πλήθος των εξωτερικών διακοπών INT1 από 0 έως 15 (Όταν φτάσει 15 να αρχίζει ξανά από το μηδέν).

Όσο είναι πατημένο το μπουτόν **PD7** (λογικό 0) η μέτρηση των διακοπών παγώνει. Όταν το μπουτόν **PD7** αφαιρεθεί ξανά, η μέτρηση συνεχίζεται από το σημείο που είχε μείνει.

Η μέτρηση θα γίνεται εντός της Ρουτίνας εξυπηρέτησης της Διακοπής INT1. Επίσης, κάθε φορά που εκτελείται αυτή η ρουτίνα θα παράγει μια χρονική καθυστέρηση 100 mS και η αντίστοιχη σημαία διακοπής θα μηδενίζεται.

Στην κύρια ρουτίνα, το πλήθος των εξωτερικών διακοπών θα απεικονίζεται, σε δυαδική μορφή, στα led **PC3-PC0**.

### Ζήτημα 2.3

Να υλοποιηθεί αυτοματισμός που να ελέγχει το άναμμα και το σβήσιμο ενός φωτιστικού σώματος. Όταν πατάμε το push button **PD3** (δηλαδή με την ενεργοποίηση της **INT1**) να ανάβει το led **PB0** της θύρας **PORTB** (που υποθέτουμε ότι αντιπροσωπεύει το φωτιστικό σώμα). Το led θα σβήνει μετά από 3 sec, εκτός και αν ενδιάμεσα υπάρξει νέο πάτημα του PD3, οπότε και ο χρόνος των 3 sec θα ανανεώνεται. Κάθε φορά που γίνεται ανανέωση να ανάβουν όλα τα led της θύρας **PORTB** (PB7-PB0) για 1 sec, μετά να σβήνουν εκτός από το led **PB0** που παραμένει συνολικά για 3 sec εκτός και αν ανανεωθεί. Ο κώδικας να δοθεί σε C.

Υποδείξεις:

1. Στην κύρια ρουτίνα χρησιμοποιήστε τη ρουτίνα `_delay_ms(1)` και έναν μετρητή που θα μετράει από το 0 έως το 3000.
2. Στη ρουτίνα υπηρεσίας διακοπής, ο μετρητής θα μηδενίζεται.
3. Επίσης στη ρουτίνα υπηρεσίας διακοπής η αντίστοιχη σημαία διακοπής θα μηδενίζεται.