Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΑΘΗΝΑ, 17/10/2025

2η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"
Χρήση εξωτερικών διακοπών στον Μικροελεγκτή AVR

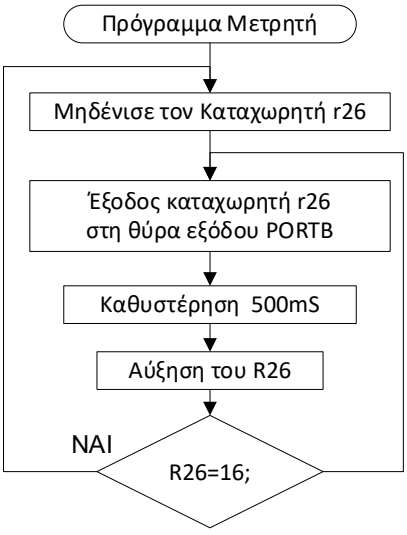
Εξέταση – Επίδειξη: Παρασκευή 24/10/2025
Προθεσμία για παράδοση Έκθεσης: Τρίτη 28/10/2025 (23:59)

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 συνδέεται 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.
Στην συνέχεια δίνεται ένα παράδειγμα προγράμματος μετρητή που διακόπτεται από την εξωτερική διακοπή INT0. Η διακοπή αυτή είναι συνδεδεμένη στο 3ο bit της PORTD (PD2).

Παράδειγμα 2.1 Ένα πρόγραμμα μέτρησης που διακόπτεται από την INT0

Αρχικά εισάγουμε ένα πρόγραμμα που μετράει δυαδικά από το 0 έως το 15 σε συνεχόμενη λειτουργία με καθυστέρηση 500 mS σε κάθε μέτρηση. Το αποτέλεσμα της μέτρησης εμφανίζεται στα Leds της θύρας PB3-PB0. Το διάγραμμα ροής του φαίνεται στο σχήμα 2.1α.



Σχήμα 2.1α Πρόγραμμα μετρητή.

Το πρόγραμμα φαίνεται στο σχήμα 2.1β παρακάτω:

Σχήμα 2.1β Πρόγραμμα μετρητή

```
.include "m328PBdef.inc"     ;ATmega328P microcontroller definitions


.equ FOSC_MHZ=16             ; Microcontroller operating frequency in MHz

.equ DEL_mS=500              ; Delay in mS (valid number from 1 to 4095)

.equ DEL_NU=FOSC_MHZ*DEL_mS ; delay_mS routine: (1000*DEL_NU+6) cycles

;Init Stack Pointer
    ldi r24, LOW(RAMEND)
    out SPL, r24
    ldi r24, HIGH(RAMEND)
    out SPH, r24

;Init PORTB as output
    ser r26
    out DDRB , r26

loop1:
    clr r26
loop2:
    out PORTB, r26

    ldi r24, low(DEL_NU)     ;
    ldi r25, high(DEL_NU)    ; Set delay (number of cycles)
    rcall delay_mS           ;

    inc r26

    cpi r26, 16              ; compare r26 with 16
    breq loop1
    rjmp loop2

; delay of 1000*F1+6 cycles (almost equal to 1000*F1 cycles)
delay_mS:

; total delay of next 4 insruction group = 1+(249*4-1) = 996 cycles
    ldi r23, 249             ; (1 cycle)
loop_inn:
    dec r23                  ; 1 cycle
    nop                      ; 1 cycle
    brne loop_inn            ; 1 or 2 cycles

    sbiw r24 ,1              ; 2 cycles
    brne delay_mS            ; 1 or 2 cycles

    ret                      ; 4 cycles
```

Εξωτερικές διακοπές στον ATmega328PB

Για τη λειτουργία του συστήματος διακοπών κάθε Μικροελεγκτή είναι απαραίτητη αρχικά η ενεργοποίηση σημαιών 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:

EIMSK:							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.

EIFR:							INTF1	INTF0

Επίσης, απαραίτητη είναι και η εκτέλεση της εντολής sei, που επιτρέπει γενικώς την πρόκληση 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.

Η ρουτίνα εξυπηρέτησης της διακοπής πρέπει να δηλωθεί στην κατάλληλη διεύθυνση του προγράμματος. Το διάνυσμα της διακοπής INT0 είναι 0x002 και το διάνυσμα της διακοπής INT1 είναι 0x004.

Για παράδειγμα, το παρακάτω τμήμα κώδικα (Σχήμα 2.2) ενεργοποιεί διακοπές στην είσοδο INT0 κατά την ανερχόμενη ακμή. 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.

Στη ρουτίνα εξυπηρέτησης της εξωτερική διακοπής INT0, ανάβουν τα led της θύρας PORTB για 500mS και στην συνέχεια συνεχίζεται η μέτρηση από το σημείο που είχε μείνει.
Η ενεργοποίηση της διακοπής INT0 δίνεται στο Σχήμα 2.2 και η ρουτίνα εξυπηρέτησης διακοπής στο Σχήμα 2.3.

Σχήμα 2.2 Εντολές που ενεργοποιούν τη διακοπή INT0

```
.org 0x0
rjmp reset
.org 0x2
rjmp ISR0

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

;Enable the INT0 interrupt(PD2)
ldi r24, (1 << INT0)
out EIMSK, r24

sei                          ;Sets the Global Interrupt Flag
```

Σχήμα 2.3 Ρουτίνα εξυπηρέτησης διακοπής

```
ISR0:
    push r25                ;
    push r24                ;
    in r24 , SREG           ; Save r24, r25, SREG
    push r24                ;

    ser r24
    out PORTB , r24

    ldi r24, low(16*500)    ;
    ldi r25, high(16*500)   ; Set delay (number of cycles)
    rcall delay_mS


    pop r24                 ;
    out SREG , r24          ; Restore r24, r25, SREG
    pop r24                 ;
    pop r25                 ;

    reti
```

Το Φαινόμενο της Αναπήδησης (ή Σπινθηρισμού)

Εάν ο χρόνος που διαρκεί η εκτέλεση μιας ρουτίνας εξυπηρέτησης διακοπής είναι πολύ μικρός, ενδέχεται να εμφανιστεί το φαινόμενο της αναπήδησης, λόγω σπινθηρισμού στον πιεστικό διακόπτη που προκαλεί την εξωτερική διακοπή. Συγκεκριμένα, ένας πιεστικός διακόπτης μπορεί να δώσει περισσότερα του ενός σήματα διακοπής τόσο κατά την πίεση του όσο και κατά την απελευθέρωση του, οπότε μετά την ολοκλήρωση της ρουτίνας εξυπηρέτησης της διακοπής, μπορεί να καταφθάσουν και άλλα σήματα διακοπής που οφείλονται στο ίδιο πάτημα ή απελευθέρωση του πιεστικού διακόπτη. Μια λύση για το πρόβλημα αυτό για τον μικροελεγκτή ATmega328PB είναι ο έλεγχος του καταχωρητή EIFR μέσα στη ρουτίνα εξυπηρέτησης διακοπής.

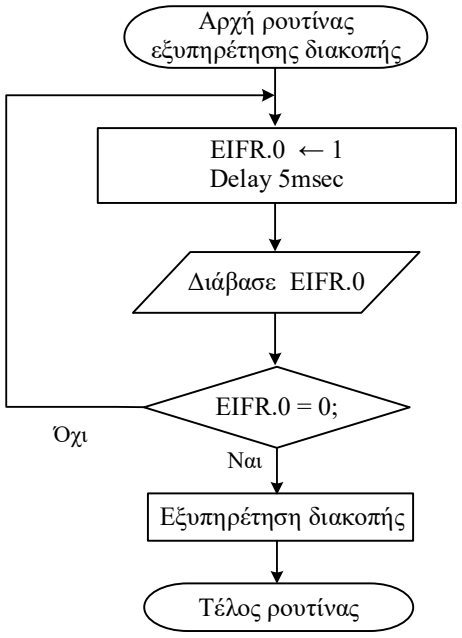
Τα ψηφία INTF1 και INTF0 του καταχωρητή EIFR τίθενται αυτόματα από το μικροελεγκτή όταν υπάρχει αίτηση εξωτερικής διακοπής INT1 και INT0 αντίστοιχα, και μηδενίζονται όταν εξυπηρετηθεί η αίτηση. Τα ψηφία INTF1 και INTF0 μπορούν να μηδενιστούν και από τον κώδικα, με εγγραφή λογικού 1 στο αντίστοιχο ψηφίο.

Π.χ. το INT0 μπορεί να μηδενιστεί με τις παρακάτω εντολές:

```
ldi r24, (1 << INTF0)
out EIFR, r24
```

Ένας απλός αλγόριθμος αποφυγής της αναπήδησης είναι ο εξής:
Στην αρχή της ρουτίνας εξυπηρέτησης διακοπής μηδενίζεται το ψηφίο INTF1 ή INTF0. Μετά από ένα μικρό χρονικό διάστημα (π.χ. 5 msec) το ψηφίο ελέγχεται ξανά και αν είναι λογικό 1 τότε κάποιος σπινθηρισμός έχει δώσει νέο σήμα διακοπής για το ίδιο πάτημα και το ψηφίο μηδενίζεται ξανά. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου το ψηφίο σταθεροποιηθεί σε λογικό 0, οπότε έχουν σταματήσει οι αναπηδήσεις, και ολοκληρώνεται η ρουτίνα εξυπηρέτησης.

Τα παραπάνω, για την περίπτωση της εξωτερικής εισόδου INT0 φαίνονται και στο επόμενο λογικό διάγραμμα.



Σχήμα 2.4 Λογικό διάγραμμα αποφυγής σπινθηρισμού (debouncing) πλήκτρου διακοπής INT0.

Παράδειγμα 2.2 Ένα πρόγραμμα με εξωτερικά Interrupts σε γλώσσα C

Το πρόγραμμα που φαίνεται στο παρακάτω σχήμα (Σχήμα 2.5) 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. Οι ρουτίνες εξυπηρέτησης των διακοπών ανάβουν τα led της PORTC για 2 Sec.

Σχήμα 2.5 Κώδικας που αναβοσβήνει τα led της PORTB και διακόπτεται από τα INT0 και 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;
    _delay_ms(2000);
    PORTC = 0x00;
}

ISR (INT1_vect)        //External INT1 ISR
{
    PORTC = 0xFF;
    _delay_ms(2000);
    PORTC = 0x00;
}

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

    while(1)
    {
    PORTB = 0x00;
    _delay_ms(500);
    PORTB = 0xFF;
    _delay_ms(500);
    }
}
```

Τα ζητούμενα της 2ης εργαστηριακής άσκησης

Ζήτημα 2.1

Α) Στον κώδικα του μετρητή του σχήματος 2.1, να προστεθεί ρουτίνα εξυπηρέτησης της εξωτερικής διακοπής **INT1** (PD3), σε γλώσσα Assembly, η οποία να απαριθμεί το πλήθος των διακοπών INT1 από 0 έως 31. Όταν φτάσει 31 να αρχίζει ξανά από το μηδέν. Όσο είναι πατημένο το μπουτόν **PD1** (λογικό 0) η μέτρηση των διακοπών παγώνει. Όταν το μπουτόν **PD1** αφεθεί ξανά, η μέτρηση συνεχίζεται από το σημείο που είχε μείνει. Το πλήθος των εξωτερικών διακοπών να απεικονίζεται, σε δυαδική μορφή, στα leds **PC5-PC1.**

Β) Επειδή υπάρχει η πιθανότητα να εμφανιστεί το φαινόμενο του σπινθηρισμού, με το πάτημα του μπουτόν **PD3** του προηγούμενου παραδείγματος, να προστεθούν οι κατάλληλες εντολές στη ρουτίνα εξυπηρέτησης της εξωτερικής διακοπής **INT1,** ώστε να αποφευχθεί το φαινόμενο αυτό.

Ζήτημα 2.2

Α) Να υλοποιηθεί κώδικας Assembly για το μικροελεγκτή ATmega328PB, αντίστοιχο με τον κώδικα του μετρητή του σχήματος 2.1, με τη διαφορά ότι η μέτρηση θα είναι 0 έως 31, η απεικόνιση θα γίνεται στα leds **PC5-PC1** και ο χρόνος καθυστέρησης μεταξύ των εναλλαγών θα κυμαίνεται στα 1000 mS.

Β) Η ρουτίνα εξυπηρέτησης της εξωτερικής διακοπής **INT0** (PD2) να τροποποιηθεί έτσι ώστε όταν ενεργοποιείται να ανάβει τόσα LEDs της θύρας **PORTC** αρχίζοντας από το **LSB** όσο το πλήθος των 4 μπουτόν PB4 – PB1 που είναι πατημένα.

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 PB3** της θύρας **PORTB** (που υποθέτουμε ότι αντιπροσωπεύει το φωτιστικό σώμα). Το led θα σβήνει μετά από 4 sec, εκτός και αν ενδιάμεσα υπάρξει νέο πάτημα του **PD3**, οπότε και ο χρόνος των 4 sec θα ανανεώνεται. Κάθε φορά που γίνεται ανανέωση να ανάβουν τα led της θύρας **PORTB** (PB5-PB0) για 1 sec, μετά να σβήνουν εκτός από το led **PB3** που παραμένει συνολικά για 4 sec εκτός και αν ανανεωθεί. Ο κώδικας να δοθεί σε **Assembly** και σε **C**.