



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΪΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΑΘΗΝΑ 1 Ιουνίου 2023

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

Εξέταση – Επίδειξη: Παρασκευή 9/5/2023.

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 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 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.

Παράδειγμα κώδικα χειρισμού εξωτερικών διακοπών

Για παράδειγμα, ο κώδικας στην επόμενη σελίδα ενεργοποιεί τις διακοπές στην είσοδο INT0 κατά την ανερχόμενη ακμή.

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.

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.

```

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

reset:

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

; 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
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

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

```

Παράδειγμα 2.2 Ένα πρόγραμμα με εξωτερικά Interrupts σε γλώσσα C

Το πρόγραμμα που φαίνεται στο παρακάτω σχήμα 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.

```
#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 2^{ης} εργαστηριακής άσκησης

Ζήτημα 2.1

Να υλοποιηθεί κώδικας Assembly για το μικροελεγκτή ATmega328PB, που απαριθμεί το πλήθος των εξωτερικών διακοπών INT1 από 0 έως 31. Όταν φτάσει 31 να αρχίζει ξανά από το μηδέν. Όσο είναι πατημένο το μπουτόν **PD7** (λογικό 0) η μέτρηση των διακοπών παγώνει. Όταν το μπουτόν **PD7** αφεθεί ξανά, η μέτρηση συνεχίζεται από το σημείο που είχε μείνει. Το πλήθος των εξωτερικών διακοπών να απεικονίζεται, σε δυαδική μορφή, στα leds **PC4-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 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 όλα τα led της θύρας **PORTB** (PB7-PB0) για 0.5 sec, μετά να σβήνουν εκτός από το led **PB0** που παραμένει συνολικά για 4 sec εκτός και αν ανανεωθεί. Ο κώδικας να δοθεί σε **C**.