ΕΜΠ - ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ & ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΑΘΗΝΑ 20 Νοεμβρίου 2015

Εργαστηριακές ασκήσεις στον Μικροελεγκτή AVR

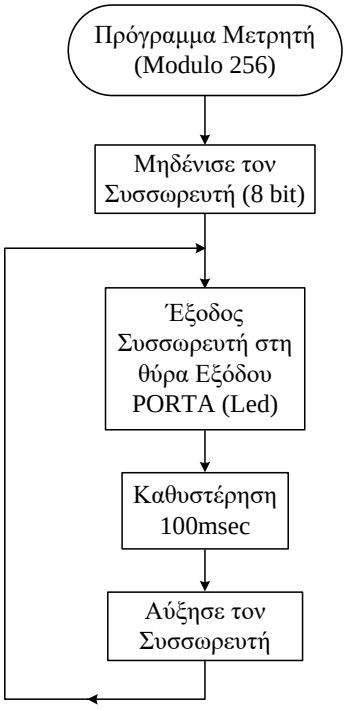
2η Εργαστηριακή Άσκηση του AVR– Χρήση εξωτερικών διακοπών

Μια χρήσιμη δυνατότητα των μικροελεγκτών είναι η άμεση ανταπόκρισή τους σε εξωτερικές συνθήκες. Η ανταπόκριση αυτή επιτυγχάνεται με την εκμετάλλευση του συστήματος διακοπών του μικροελεγκτή. Ως γνωστόν, κάθε μικροελεγκτής είναι εφοδιασμένος με μια ή περισσότερες εισόδους διακοπών. Η ενεργοποίηση μια εισόδου διακοπής υποχρεώνει το μικροελεγκτή να σταματήσει άμεσα όποια εργασία κάνει και να εκτελέσει τον κώδικα που υπάρχει σε μια προκαθορισμένη διεύθυνση, που ονομάζεται και διάνυσμα διακοπής. Στο σημείο αυτό συνδέεται συνήθως μια ρουτίνα εξυπηρέτησης διακοπής (διαφορετική για κάθε εφαρμογή). Με το τέλος της ρουτίνας εξυπηρέτησης διακοπής, ο μικροελεγκτής συνεχίζει την εργασία που διέκοψε, από επιστρέφοντας στο σημείο ακριβώς που είχε διακοπεί.

Στην άσκηση αυτή θα γίνει μελέτη της χρήσης των διακοπών του AVR στο σύστημα easyAVR6. 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 255 και ξαναρχίζει σε συνεχόμενη λειτουργία με καθυστέρηση 100 ms σε κάθε μέτρηση. Το αποτέλεσμα της μέτρησης εμφανίζεται στα Led της θύρας PORTA. Το διάγραμμα ροής του φαίνεται στο σχήμα 2.1.



Σχήμα 2. 1. Πρόγραμμα μετρητή.

Υποθέτουμε ότι όταν προκαλείται εξωτερική διακοπή INT0, ανάβουν όλα τα led της θύρας PORTA για 1sec και στην συνέχεια επανέρχεται η μέτρηση (από το σημείο που είχε μείνει). Το listing του προγράμματος φαίνεται στον πίνακα 2.1 και η ενεργοποίηση της διακοπής INT0 (με αποκλεισμό όλων των άλλων) δίνεται στον πίνακα 2.2. Η ρουτίνα εξυπηρέτησης διακοπής της ζητούμενης λειτουργίας δίνεται στον πίνακα 2.3.

Πίνακας 2.1. Πρόγραμμα μετρητής

Ετικέτα	Εντολή	Σχόλια
	ser r26	; αρχικοποίηση της PORTA
	out DDRA , r26	; για έξοδο
	clr r26	; αρχικοποίηση του μετρητή
loop:	out PORTA , r26	; Δείξε την τιμή του μετρητή
		; στη θύρα εξόδου των LED
	ldi r24 , low(100)	; load r25:r24 with 100
	ldi r25 , high(100)	; delay 100 ms
	rcall wait_msec	
	inc r26	; Αύξησε τον μετρητή
	rjmp loop	; Επανέλαβε

Για τη λειτουργία του συστήματος διακοπών κάθε μικροελεγκτή είναι απαραίτητη αρχικά η ενεργοποίηση σημαιών και επιλογών, που καθορίζουν τον ακριβή τρόπο λειτουργίας του. Στον μικροελεγκτή AVR ATmega16 οι δύο βασικές σημαίες είναι η επιλογή του επιπέδου ενεργοποίησης διακοπής και η επίτρεψη της επιθυμητής εισόδου διακοπής. Η πρώτη ενεργοποιείται γράφοντας στον καταχωρητή MCUCR (διεύθυνση $35) και στα τέσσερα λιγότερα σημαντικά ψηφία κατάλληλες τιμές, σύμφωνα με το παρακάτω σχήμα και τους παρακάτω πίνακες:

MCUCR:	SRE	SRW	SE	SM	**ISC11**	**ISC10**	**ISC01**	**ISC00**

ISC11	ISC10	Περιγραφή
0	0	Διακοπή στη χαμηλή στάθμη του INT1
0	1	Δεσμευμένο
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	Δεσμευμένο
1	0	Διακοπή στην κατερχόμενη ακμή του INT0
1	1	Διακοπή στην ανερχόμενη ακμή του INT0

Η δεύτερη ενεργοποιείται γράφοντας στον καταχωρητή GICR (διεύθυνση $3B) την τιμή 1 στο ψηφίο που αντιστοιχεί στην είσοδο διακοπής που επιθυμούμε να επιτρέψουμε, σύμφωνα με το παρακάτω σχήμα:

GICR:	INT1	INT0						

Επίσης, απαραίτητη είναι και η εκτέλεση της εντολής sei, που επιτρέπει την πρόκληση διακοπών. Για παράδειγμα, το παρακάτω τμήμα κώδικα ενεργοποιεί διακοπές στην είσοδο INT0 κατά την ανερχόμενη ακμή. Τέλος, θα πρέπει οι ρουτίνες εξυπηρέτησης της αντίστοιχης διακοπής να δηλωθεί στην κατάλληλη διεύθυνση του προγράμματος (οι διευθύνσεις περιγράφονται λεπτομερώς στο βιβλίο), όπως στο παρακάτω παράδειγμα. Υπενθυμίζεται ότι στον μικροελεγκτή AVR ATmega16 η είσοδος PD2 λειτουργεί εναλλακτικά και ως εξωτερική είσοδος διακοπής INT0 και η PD3 ως INT1. Επίσης, **η διεύθυνση της INT0 είναι η 0x2 και η διεύθυνση της INT1 η 0x4.**

Πίνακας 2.2. Εντολές που ενεργοποιούν τη διακοπή INT0 (και εφαρμόζει μάσκα σε όλες τις άλλες)

Εντολή	Σχόλια
.org 0x0	; Η αρχή του κώδικα (reset) πάντα
rjmp reset	; θα δηλώνεται στην δ/ση 0x0
.org 0x2	; Η εξυπηρέτηση της INT0
rjmp ISR0	; ορίζεται στην δ/ση 0x2
reset:	
ldi r24 ,(1 << ISC01) | (1 << ISC00)	; Εδώ αρχίζει το κυρίως τμήμα
	; του προγράμματος.
	; ορίζεται η διακοπή `INT0` να
out MCUCR , r24	; προκαλείται με σήμα θετικής ακμής
ldi r24 ,(1 << INT0)	; Ενεργοποίησε τη διακοπή `INT0`
out GICR , r24	
sei	; Ενεργοποίησε τις συνολικές διακοπές

Πίνακας 2.3. Ρουτίνα εξυπηρέτησης διακοπής

Ετικέτα	Εντολή	Σχόλια
ISR0:	push r26	; Σώσε το περιεχόμενο των `r26`
	in r26 , SREG	; και `SREG`
	push r26	
	ser r26	; θέσε τη θύρα εξόδου των LED
	out PORTA , r26	
	ldi r24 , low(998)	; φόρτωσε τους r25:r24 με 980
	ldi r25 , high(998)	; delay 1sec
	rcall wait_msec	
	out SREG , r26	; καταχωρητών `r24` και `SREG`
	pop r26	
	reti	; Επιστροφή από διακοπή στο κύριο πρόγραμμα

Τα ζητούμενα της 2ης εργαστηριακής άσκησης του AVR

Ζήτημα 2.1 Υποθέτουμε ότι «τρέχει» το προηγούμενο πρόγραμμα του μετρητή. Να δοθεί ρουτίνα εξυπηρέτησης της εξωτερικής διακοπής INT1 (PD3) που όταν ενεργοποιείται να απαριθμεί το πλήθος τους με την προϋπόθεση ότι το dip switch PD7 είναι στο λογικό ‘1’, αλλιώς όχι. Ο μετρητής που αποτελεί το κύριο πρόγραμμα να απεικονίζεται στα leds PB7-PB0 και να τρέχει με ταχύτητα μιας μέτρησης ανά ~ 0.2 του δευτερολέπτου. Η μέτρηση του πλήθους των εξωτερικών διακοπών INT1 να δίνεται στα leds PA7-PA0. Το πρόγραμμα να δοθεί σε assembly.

Ζήτημα 2.2 Στο προηγούμενο πρόγραμμα του μετρητή, να δοθεί μια άλλη ρουτίνα εξυπηρέτησης της εξωτερικής διακοπής INT0 (PD2) που όταν ενεργοποιείται να ανάβει τόσα LEDs της θύρας PORTC (PC0-7) αρχίζοντας από τα LSB όσο το πλήθος των διακοπτών (dip switches) της θύρας PORTA (PA7-PA0) που είναι ON. Το πρόγραμμα να δοθεί σε assembly.

Το Φαινόμενο της Αναπήδησης (Σπινθηρισμού)

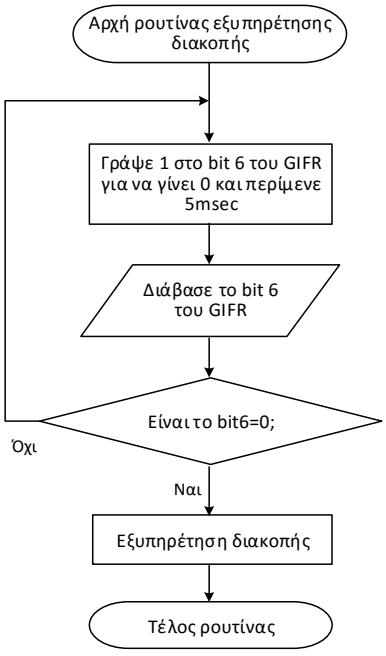
Εάν ο χρόνος που διαρκεί η εκτέλεση μιας ρουτίνας εξυπηρέτησης διακοπής είναι πολύ μικρός, ενδέχεται να εμφανιστεί το φαινόμενο της αναπήδησης λόγω σπινθηρισμού στον πιεστικό διακόπτη που προκαλεί την εξωτερική διακοπή. Συγκεκριμένα, ένα πάτημα του πιεστικού διακόπτη μπορεί να δώσει περισσότερα του ενός σήματα διακοπής τόσο κατά την πίεση όσο και κατά την απελευθέρωση. Εφόσον η ρουτίνα εξυπηρέτησης διακοπής διαρκεί λίγο, ένας σπινθηρισμός μπορεί να προλάβει να εξυπηρετηθεί πλήρως και στη συνέχεια να καταφθάσουν και άλλα σήματα διακοπής, από το ίδιο πάτημα ή το άφημα του πιεστικού διακόπτη. Το αποτέλεσμα θα είναι να προκληθούν νέες αιτήσεις διακοπής οι οποίες και θα εξυπηρετηθούν. Όπως αναφέρθηκε, τα push buttons εμφανίζουν αναπήδηση και όταν αφήνονται, με αποτέλεσμα να υπάρχει η πιθανότητα σε ένα πάτημα του κουμπιού να εκτελεστεί δύο φορές η ρουτίνα εξυπηρέτησης διακοπής (μια φορά όταν το πατάμε και μια όταν το αφήνουμε). Μια λύση για το πρόβλημα αυτό για τον μικροελεγκτή AVR ATmega16 είναι ο έλεγχος από τη ρουτίνα εξυπηρέτησης διακοπής του καταχωρητή GIFR (διεύθυνση $3A) και συγκεκριμένα των δύο περισσότερο σημαντικών ψηφίων, σύμφωνα με το παρακάτω σχήμα.

GIFR:	INTF1	INTF0						

Τα ψηφία INTF1 και INTF0 γίνονται λογικό 1 από το μικροελεγκτή όταν υπάρχει αίτηση εξωτερικής διακοπής INT1 και INT0 αντίστοιχα. Όταν εξυπηρετηθεί η αίτηση γίνονται λογικό 0. Αυτό το μηδενισμό μπορεί όμως να τον κάνει και ο χρήστης, γράφοντας λογικό 1 (προσοχή!) στο αντίστοιχο ψηφίο, π.χ. με τις παρακάτω εντολές (για την περίπτωση του INT0).

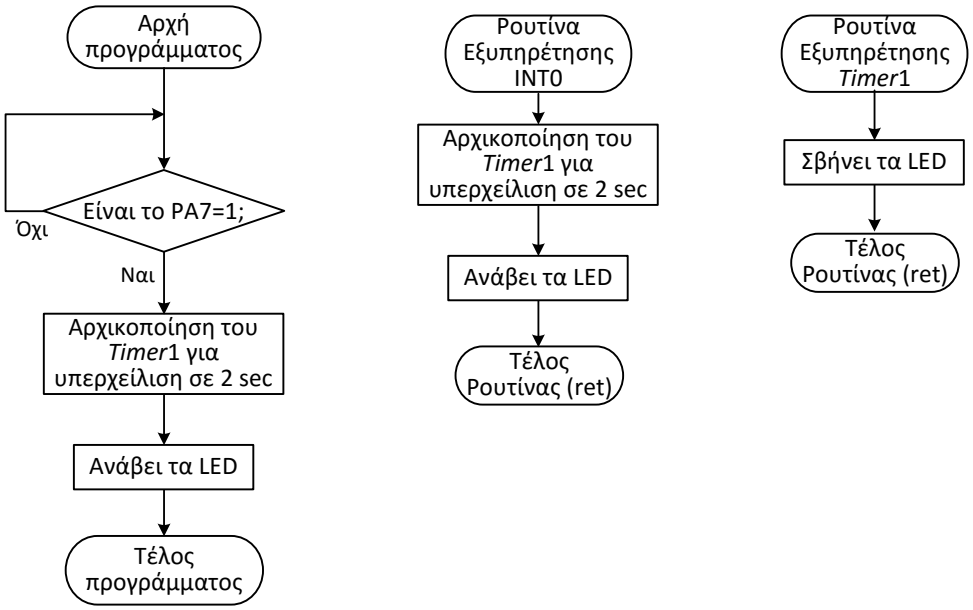
```
ldi r24 ,(1 << INTF0)
out GIFR ,r24                ; μηδένισε το bit 6 του GIFR
```

Ένας λοιπόν απλός αλγόριθμος αποφυγής της αναπήδησης είναι ο χρήστης να μηδενίζει στην αρχή της ρουτίνας εξυπηρέτησης διακοπής το ψηφίο INTF1 ή INTF0 και να περιμένει ένα μικρό χρονικό διάστημα (π.χ. 5msec). Στη συνέχεια το ελέγχει και αν είναι πάλι λογικό 1, τότε κάποιος σπινθηρισμός έχει δώσει νέο σήμα διακοπής για το ίδιο πάτημα. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου το ψηφίο ισορροπήσει στην τιμή 0 που επιβάλει ο χρήστης, οπότε έχουν σταματήσει οι αναπηδήσεις και μπορεί να εξυπηρετηθεί η μία και μοναδική διακοπή. Τα παραπάνω, για την περίπτωση της εξωτερικής εισόδου INT0 φαίνονται και στο επόμενο λογικό διάγραμμα.



Σχήμα 2. 2. Λογικό διάγραμμα αποσπινθηρισμού (debouncing) πλήκτρου διακοπής INT0.

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) ή το PA7 (που υποθέτουμε ότι αντιστοιχεί σε ένα αισθητήρα κίνησης) να ανάβει το led PB0 (που αντιπροσωπεύει το φωτιστικό σώμα). Το led θα σβήνει μετά από ~ 2 sec, εκτός και αν ενδιάμεσα υπάρξει νέο πάτημα του PD3 ή PA7, οπότε και ο χρόνος των 2 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 2 sec εκτός και αν ανανεωθεί. Να γίνει χρήση του χρονιστή *Timer*1. Μπορείτε να βασιστείτε στο Διάγραμμα ροής που δίνεται στο Σχ. 2.3. Περιγράψτε επίσης την συνολική λειτουργία.



Σχήμα 2. 3. Λογικό διάγραμμα αυτοματισμού που ανάβει για 2 sec το led.

Χρονιστής *Timer*1 (επαναλαμβάνεται από την θεωρία για την πληρότητα της άσκησης)

Ένα πλεονέκτημα των σύγχρονων μικροελεγκτών είναι η ενσωμάτωση στην ίδια ψηφίδα χρήσιμων περιφερειακών συσκευών, με σύνδεση στο σύστημα διακοπών του μικροελεγκτή. Για παράδειγμα, οι χρονιστές είναι καταχωρητές που μπορούν να προγραμματιστούν να προκαλέσουν διακοπή στον μικροελεγκτή μετά την πάροδο συγκεκριμένου χρόνου.

Ο μικροελεγκτής AVR ATmega16 διαθέτει 2 χρονιστές. Τον 8-ψήφιο και απλούστερο TCNT0 (διεύθυνση $32) και τον 16-ψήφιο TCNT1 (TCNT1H, διεύθυνση $2D και TCNT1L, διεύθυνση $2C). Στους χρονιστές αυτούς μπορεί να τοποθετηθεί μια αρχική τιμή η οποία αυξάνεται από το μικροελεγκτή με επιλεγμένη συχνότητα. Όταν ένας χρονιστής υπερχειλίσει, μπορεί να δημιουργήσει κατάλληλο σήμα διακοπής εφόσον τροποποιηθεί το περιεχόμενο του καταχωρητή TIMSK (διεύθυνση $39) σύμφωνα με το παρακάτω σχήμα. Γράφοντας 1 στο ψηφίο TOIE0 επιτρέπονται διακοπές υπερχείλισης του χρονιστή TCNT0 και αντίστοιχα με 1 στο ψηφίο TOIE1 επιτρέπονται διακοπές υπερχείλισης του χρονιστή TCNT1 (εφόσον επιτραπούν και γενικά οι διακοπές με την εντολή sei).

TIMSK:								
						TOIE1		TOIE0

Για παράδειγμα, με τις παρακάτω εντολές επιτρέπεται η διακοπή υπερχείλισης του μετρητή TCNT1.

```
ldi r24 ,(1<<TOIE1)      ; ενεργοποίηση διακοπής υπερχείλισης του μετρητή TCNT1
out TIMSK ,r24            ; για τον timer1
```

Η επιλογή συχνότητας αύξησης του κάθε χρονιστή γίνεται διαφορετικά. Για το χρονιστή TCNT1 χρησιμοποιούμε τον καταχωρητή TCCR1B (διεύθυνση $2E) σύμφωνα με το παρακάτω σχήμα και τον παρακάτω πίνακα.

TCCR1B:						CS12	CS11	CS10

CS12	**CS11**	**CS10**	**Περιγραφή σήματος εισόδου χρονιστή**
0	0	0	Κανένα. Χρονιστής σταματημένος
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	Κατερχόμενη ακμή εξωτερικού σήματος ακροδέκτη T1
1	1	1	Ανερχόμενη ακμή εξωτερικού σήματος ακροδέκτη T1

Για παράδειγμα, με τις παρακάτω εντολές επιλέγεται συχνότητα αύξησης του χρονιστή TCNT1 ίση με τη 1/1024 της συχνότητας ρολογιού του μικροελεγκτή (δηλαδή, κάθε 1024 κύκλους ρολογιού αυξάνεται κατά 1 ο χρονιστής TCNT1).

```
ldi r24 ,(1<<CS12) | (0<<CS11) | (1<<CS10) ; CK/1024
out TCCR1B ,r24
```

Στην αναπτυξιακή πλακέτας EasyAVR6 με συχνότητα ρολογιού 8MHz, η επιλογή αυτή ισοδυναμεί με συχνότητα αύξησης του TCNT1 ίση με 8MHz/1024=7812.5Hz. Αν με αυτές τις επιλογές θέλουμε ο TCNT1 να δημιουργήσει σήμα διακοπής υπερχείλισης μετά από για παράδειγμα 5sec, πρέπει να τον κάνουμε να μετρήσει 5×7812.5=39062.5 κύκλους. Επειδή η υπερχείλιση γίνεται όταν μετρήσει 65536 κύκλους (16 ψηφία), θα πρέπει η αρχική τιμή που θα του δοθεί πριν αρχίσει να μετράει προς τα πάνω να είναι 65536-39062.5=26473.5=0x6769. Αυτό γίνεται με τον παρακάτω κώδικα.

```
ldi r24,0x67            ; αρχικοποίηση του TCNT1 to
out TCNT1H ,r24         ; για υπερχείλιση μετά από 5 sec
ldi r24 ,0x69
out TCNT1L ,r24
```

Για μέγιστη ασφάλεια, τα δύο τμήματα του μετρητή TCNT1 TCNT1H και TCNT1L πρέπει να διαβάζονται αδιαίρετα, χωρίς να μεσολαβήσει για παράδειγμα κάποια άλλη διακοπή. Για το λόγο αυτό ο μικροελεγκτής κάνει μια ειδική διαδικασία. Όταν μια τιμή γράφεται στον TCNT1H αυτή τοποθετείται στον προσωρινό καταχωρητή TEMP. Στη συνέχεια, όταν γραφεί τιμή στον TCNT1L η τιμή που υπάρχει στον TEMP συνδυάζεται με αυτή και τα 16 ψηφία γράφονται ταυτόχρονα σε όλο το μήκος του TCNT1. Κατά την ανάγνωση, όταν διαβάζεται μια τιμή από τον TCNT1L αυτή τοποθετείται στον επιλεγμένο καταχωρητή του μικροελεγκτή και ταυτόχρονα η τιμή του TCNT1H μεταφέρεται στον καταχωρητή TEMP. Όταν στην συνέχεια διαβαστεί και ο TCNT1H, μεταφέρεται στον επιλεγμένο καταχωρητή η τιμή που έχει τοποθετηθεί στον TEMP. Με βάση αυτή τη διαδικασία κατά την εγγραφή πρέπει πάντα να γράφεται πρώτα η τιμή στον TCNT1H και μετά στον TCNT1L ενώ κατά την ανάγνωση πρέπει πάντα να διαβάζεται πρώτα ο TCNT1L και μετά ο TCNT1H. Σε πολύπλοκες εφαρμογές πριν την πρόσβαση στον TCNT1 μπορούν να απενεργοποιηθούν και οι διακοπές (αν και δεν συνηθίζεται). Τέλος, η θέση μνήμης του διανύσματος διακοπής του χρονιστή TCNT1 φαίνεται στον παρακάτω κώδικα.

```
.org 0x10
rjmp ISR_TIMER1_OVF  ; ρουτίνα εξυπηρέτησης της διακοπής υπερχείλισης του timer1
```