Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΑΚΑΔ. ΕΤΟΣ 2020-2021

ΑΘΗΝΑ 26 Οκτωβρίου 2020

2η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"
2η Εργ. Άσκ. στον Μικροελεγκτή AVR – Λογικές Πράξεις / Χρήση εξωτερικών διακοπών
(υλοποίηση στον προσομοιωτή Atmel Studio)

Εξέταση – Επίδειξη: Τετάρτη 4/11/2020.
Προθεσμία για παράδοση Έκθεσης: Κυριακή 8/11/2020

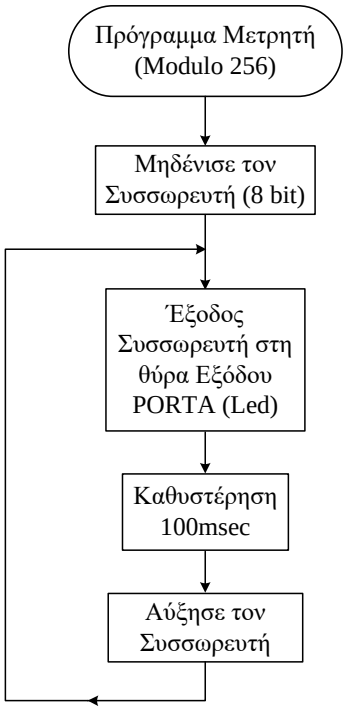
Εισαγωγή

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 στον προσομοιωτή Atmel Studio. Στην συνέχεια δίνεται ένα παράδειγμα προγράμματος μετρητή που διακόπτεται από την εξωτερική διακοπή INT0.

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. (Όταν θα εκτελείται το πρόγραμμα στις πραγματικές πλακέτες του εργαστηρίου θα χρειάζεται η χρονοκαθυστέρηση των 100msec). **Στον προσομοιωτή δεν θα χρησιμοποιείτε χρονοκαθυστερήσεις.**



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
		; ΠΡΟΣΟΧΗ: **η κλήση για χρονοκαθυστέρηση να μπαίνει σε σχόλιο για εκτέλεση στον προσομοιωτή.**
	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)

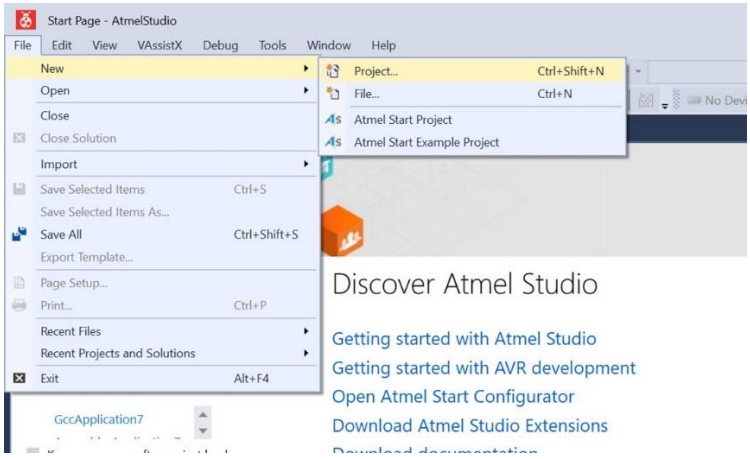
Εντολή	Σχόλια
.org 0x0	; Η αρχή του κώδικα (reset) πάντα
rjmp reset	; θα δηλώνεται στην δ/ση 0x0
.org 0x2	; Η εξυπηρέτηση της INT0
rjmp ISR0	; ορίζεται στην δ/ση 0x2
reset:	
ldi r24 ,(1 << ISC01) | (1 << ISC00)	; Εδώ αρχίζει το κυρίως τμήμα
	; του προγράμματος.
	; ορίζεται η διακοπή `INT0` να
out MCUCR , 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
		; **Όταν εκτελείτε στην πλακέτα θα κάνετε χρήση**
	rcall wait_msec	**χρονοκαθυστέρησης. Σε προσομοίωση όχι.**
	out SREG , r26	; καταχωρητών `r24` και `SREG`
	pop r26	
	reti	; Επιστροφή από διακοπή στο κύριο πρόγραμμα

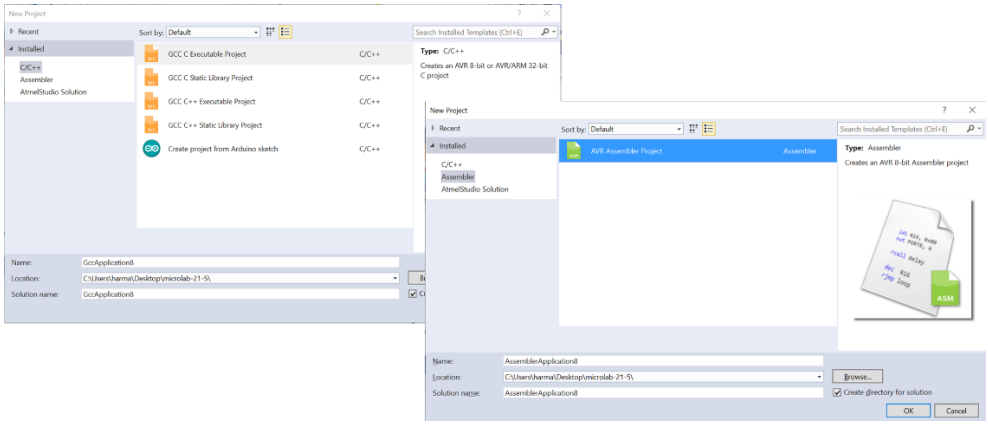
Σύντομες οδηγίες για Χρήση του Atmel Studio (με εικόνες)

1. Δημιουργία νέου project διαλέγοντας New → Project



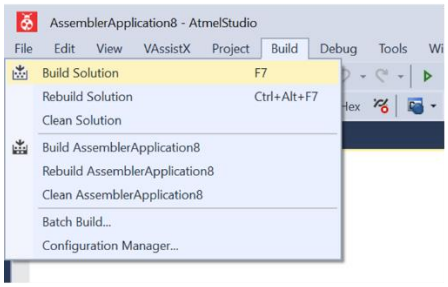
2. Δημιουργία νέου project

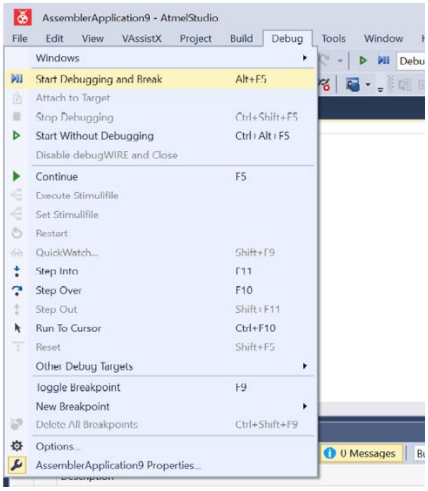
Επιλέγουμε C/C++ → GCC C Executable Project για να τρέξουμε πρόγραμμα σε C και Assembler → AVR Assembler Project για να τρέξουμε πρόγραμμα σε Assembly



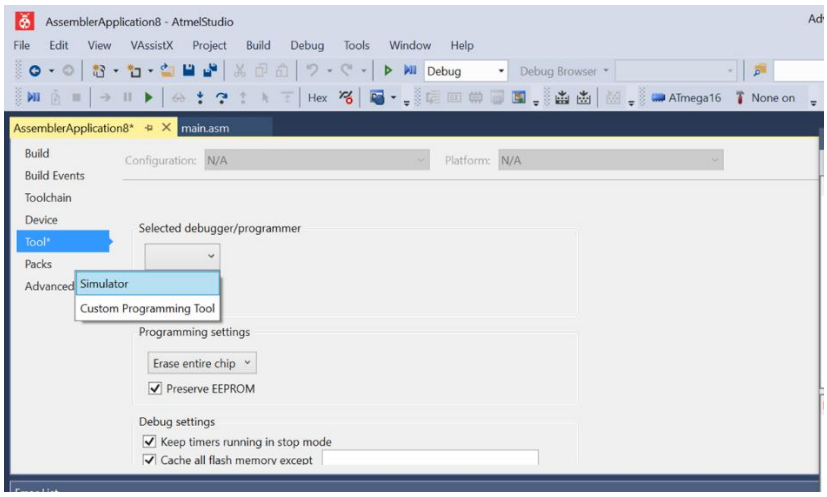
3. Build

Για να κάνουμε compile και build χρησιμοποιούμε το πλήκτρο *Start Debugging and Break*



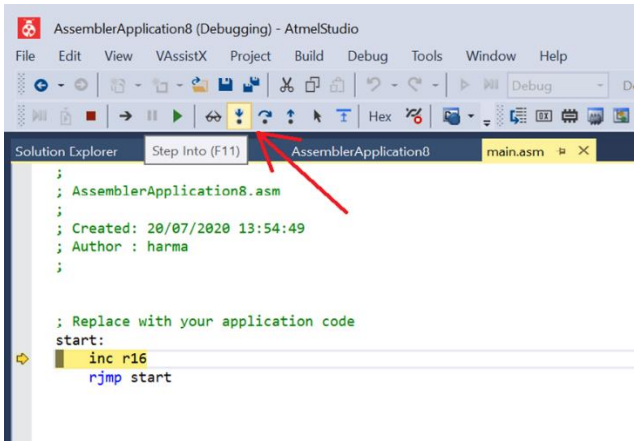


Αν βγάλει ένα παράθυρο όπως από κάτω διαλέξτε Simulator και Ctrl+S για save

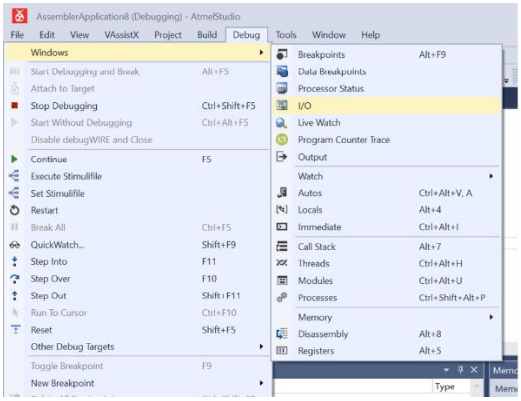


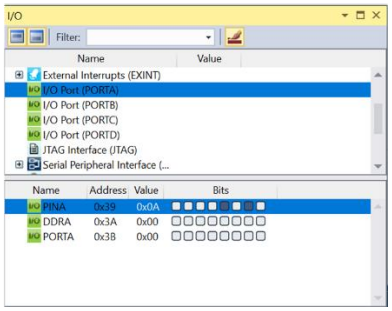
4. Έλεγχος/προσομοίωση προγράμματος

Για να ελέγχετε το πρόγραμμά σας μπορείτε να χρησιμοποιείτε βηματική εκτέλεση (εντολή εντολή) με το κουμπί που φαίνεται στην εικόνα



Αν δεν βλέπετε τα I/O πατήστε Debug → Weindows → I/O. Στο παράθυρο που βγαίνει μπορείτε να διαλέγετε το Port που σας ενδιαφέρει για είσοδο ή έξοδο. Στα PIN βάζετε την είσοδο στο PORT βλέπετε την έξοδο κάθε θύρας.





- Οι διακοπές μπορούν να προσομοιωθούν κανονικά στο Atmel Studio και να ενεργοποιούνται με πάτημα των πλήκτρων PD όπως στην πλακέτα.
- Στην προσομοίωση με χρήση breakpoints λειτουργούν κανονικά.
- Στην βηματική εκτέλεση του προγράμματος για να λειτουργήσουν θα πρέπει να θέσετε το "*Mask interrupts while stepping*" σε *False*. Η ρύθμιση αυτή βρίσκεται στο Tools → Options → και στην λίστα αριστερά στο παράθυρο που βγαίνει Tools → Tool Settings.
- Υπενθυμίζεται ότι δεν χρησιμοποιούμε χρονοκαθυστερήσεις στην προσομοίωση

Τα ζητούμενα της 2ης εργαστηριακής άσκησης (1η AVR)

Ζήτημα 2.1

Να υλοποιηθούν σε ένα σύστημα Μικροελεγκτή AVR (προσομοίωση στο Atmel Studio 7) οι λογικές συναρτήσεις:

$F0 = (A'B + B'CD)'$

$F1 = (A \cdot C) \cdot (B + D)$

Οι τιμές των *F*0- *F*1 να εμφανιστούν αντίστοιχα στα δύο LSB της θύρας εξόδου **PORTB (0-1).** Οι μεταβλητές εισόδου (*A, B, C, D*) υποθέτουμε ότι αντιστοιχούν στα 4 bit της θύρας εισόδου **PORTC (0-3)**, με το *A* στο LSB. Το πρόγραμμα να δοθεί σε **assembly και C**.

Ζήτημα 2.2

Υποθέτουμε ότι «τρέχει» το προηγούμενο πρόγραμμα του μετρητή (Πίνακας 2.1) με θύρα όμως εξόδου το **PORTC**. Να δοθεί ρουτίνα εξυπηρέτησης της εξωτερικής διακοπής **INT1** (PD3) που όταν ενεργοποιείται να απαριθμεί το πλήθος των διακοπών με την προϋπόθεση ότι τα dip switches **PA7 και PA6** είναι στο λογικό '1', αλλιώς όχι. Ο παραπάνω μετρητής (Πίνακας 2.1, με θύρα όμως εξόδου το PORTC) που αποτελεί το κύριο πρόγραμμα, να απεικονίζει στα leds **PC7-PC0** την μέτρησή του. Η μέτρηση του πλήθους των εξωτερικών διακοπών **INT1** να δίνεται σε δυαδική μορφή στα leds **PB7-PB0**. Το πρόγραμμα να δοθεί σε **assembly**.

Ζήτημα 2.3

Να δοθεί πρόγραμμα με ρουτίνα εξυπηρέτησης συνδεδεμένης με την εξωτερική διακοπή **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 **PORTB** (PB7-PB0) που είναι **ON μόνο αν το PA2** είναι **OFF**. Αν το PA2 είναι **ON** τότε κατά την ρουτίνα εξυπηρέτησης να φαίνεται στην **PORTC** το πλήθος των διακοπτών (dip switches) της θύρας **PORTB** (PB7-PB0) που είναι **ON** σε **δυαδική μορφή.** Το πρόγραμμα να δοθεί σε **C**.

ΠΡΟΣΟΧΗ το παρακάτω αφορά την εκτέλεση μόνο στην πλακέτα και όχι στον προσομοιωτή για την άσκηση που θα γίνει στις πλακέτες του εργαστηρίου:

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, 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.

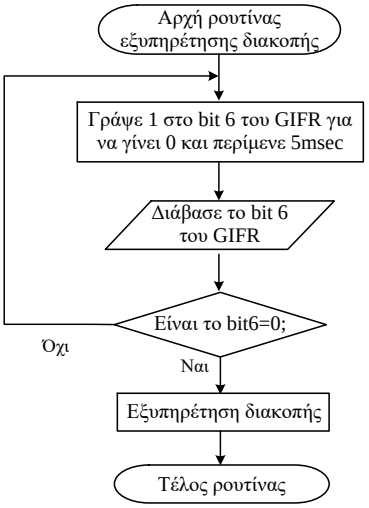
GIFR:

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. 5 msec). 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.