Ενσωματωμένα Συστήματα Μικροεπεξεργαστών – ΗΡΥ411

Αναφορά Εργαστηρίου 9 – Αφαιρώντας την αναπήδηση διακοπτών χωρίς και με Εξωτερικές Διακοπές (Interrupts)

Ομάδα: LAB41145877

Κιούλος Ευάγγελος 2016030056

Εισαγωγή:

Σκοπός της ένατης εργαστηριακής άσκησης ήταν η αφαίρεση της αναπήδησης διακοπτών με την χρήση κώδικα. Η αφαίρεση της αναπήδησης έγινε με δύο τρόπους. Ο πρώτος τρόπος, είναι με δειγματοληψία της εισόδου ανά 1 msec χρησιμοποιώντας τον TIMER/COUNTERO(Polling) και ο δεύτερος τρόπος είναι με την χρήση των εξωτερικών Interrupts. Στο συγκεκριμένο εργαστήριο υλοποιείται η αφαίρεση της αναπήδησης για διακόπτη SPDT(Single Pole Double Throw) με δύο ακροδέκτες Α και Α΄.

Υλοποίηση Λύσης με Polling:

Χάρτης Μνήμης:

0x0060	0x0061	0x0062	 	 	
			 	 	0x045F

- Διεύθυνση "0x0060": Παρούσα κατάσταση του διακόπτη. Αν έχει την τιμή '0' η κατάσταση είναι "STATE_BOUNCING", αν έχει την τιμή '1' η κατάσταση είναι "STATE SET".
- Διεύθυνση "0x0061": Τελευταία κατασταλαγμένη είσοδος.
- Διεύθυνση "0x0062": Αποθήκευση counter για μέτρηση του χρονικού πλαισίου των 10 msec.
- Διεύθυνση "0x045F": Stack

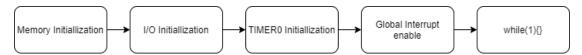
Υλοποίηση Κώδικα:

Αρχικά, δημιουργήθηκε η συνάρτηση "void TIMERO_init(void)" για την αρχικοποίηση του 8bit TIMER/COUNTERO, ο οποίος χρησιμοποιήθηκε σε Normal mode με prescaler $f_{clk}/64$ και αρχική τιμή του TCNT0 την τιμή "100". Η τιμή του prescaler και του TCNT0 υπολογίστηκε μέσο του τύπου $f_{OCnA} = \frac{f_{clk}}{2N(1+OCRnA)}$ (σελίδα 102 ATmega16 manual). Ο υπολογισμός του prescaler και του TCNTO, υπάρχει αναλυτικότερα στην αναφορά του εργαστηρίου 1. Τέλος, το bit TOIE0 του καταχωρητή TIMSK τέθηκε στην τιμή '1' για την ενεργοποίηση του TimerO Overflow Interrupt.

Στην συνέχεια, χρησιμοποιήθηκε το PIN0(PA0) του PORTA για τον ακροδέκτη A και το PIN1(PA1) του PORTA για τον ακροδέκτη A'. Η έξοδος του συστήματος ορίστηκε στο PIN2(PA2) του PORTA. Στην παρούσα υλοποίηση, ο ακροδέκτης A(PA0) χρησιμοποιείται σαν το "ON" του διακόπτη και ο ακροδέκτης A' σαν το "OFF" του διακόπτη. Επομένως, όταν η τιμή της εισόδου κατασταλάξει, στην έξοδο(PA2) θα υπάρχει η τιμή του ακροδέκτη A(PA0). Για την περιγραφή της κατάστασης του διακόπτη έχουν οριστεί δύο καταστάσεις. Η

κατάσταση "STATE_SET" στην οποία η τιμή του διακόπτη έχει κατασταλάξει και η κατάσταση "STATE_BOUNCING" στην οποία ο διακόπτης βρίσκεται σε αναπήδηση. Για την αποθήκευση της παρούσας κατάστασης, χρησιμοποιείται η διεύθυνση "0x0060" της μνήμης RAM. Επιπλέον, χρησιμοποιείται και η διεύθυνση "0x0061" για την αποθήκευση της τελευταίας κατασταλαγμένης εισόδου.

Το σώμα του κύριου προγράμματος(main()) περιγράφεται από το παρακάτω διάγραμμα:



Η λογική της αφαίρεσης της αναπήδησης, βρίσκεται στην ρουτίνα εξυπηρέτησης του Timer0 Overflow Interrupt. Αρχικά, με την χρήση του pointer "*curr_state" διαβάζουμε από την διεύθυνση "0x0060" της RAM την παρούσα κατάσταση του διακόπτη ενώ με τον pointer "*prev value" διαβάζουμε την τιμή της τελευταίας κατασταλαγμένης εισόδου. Επιπλέον, ορίζουμε τον pointer "*counter" ο οποίος χρησιμοποιείται για την μέτρηση του παραθύρου των 10 msec μετά από το οποίο θα βγάλουμε την έξοδο μας και η τιμή του αποθηκεύεται στην διεύθυνση "0x0062" της μνήμης. Στην συνέχεια διαβάζουμε την τιμή του ακροδέκτη A(PA0) μέσω της μεταβλητής "alpha" και του ακροδέκτη Α'(PA1) μέσω της μεταβλητής "alpha_bar". Με την χρήση ενός switch statement, ελέγχουμε σε ποια κατάσταση βρίσκεται ο διακόπτης. Εάν ο διακόπτης βρίσκεται στην κατάσταση "STATE SET", ελέγχουμε αν η τιμή του ακροδέκτη Α είναι η ίδια με την τελευταία σταθερή τιμή του και αν η τιμή του ακροδέκτη Α΄ είναι η αντίθετη αυτής. Ο έλεγχος αυτός γίνεται γιατί όταν η είσοδος έχει κατασταλάξει, οι ακροδέκτες Α και Α΄ έχουν αντίθετη τιμή, το οποίο προκύπτει από την κατασκευή του διακόπτη SPDT. Αν η παραπάνω συνθήκη ισχύει, τότε η κατάσταση παραμένει να είναι η "STATE_SET" και επιστρέφουμε στο κυρίως πρόγραμμα, αλλιώς, η παρούσα κατάσταση "STATE BOUNCING". Εάν ο διακόπτης βρίσκεται στην "STATE_BOUNCING", τότε ελέγχουμε την τιμή του pointer "*counter". Αν αυτή είναι μικρότερη του '10' (δηλαδή δεν έχει περάσει ο χρόνος των 10 msec) την αυξάνουμε κατά '1' και επιστρέφουμε από το interrupt. Αν είναι ίση με το '10', τότε ο χρόνος των 10 msec έχει συμπληρωθεί οπότε εμφανίζουμε την τιμή του ακροδέκτη Α(ΡΑΟ) στην έξοδο(ΡΑ2), την αποθηκεύουμε στην διεύθυνση "0x0061", θέτουμε την τιμή του δείκτη "*counter" στην τιμή '0', μεταβάλουμε την κατάσταση σε "STATE SET" και επιστρέφουμε από το interrupt.

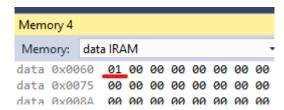
ΣΗΜΕΊΩΣΗ: Ο κώδικας της παραπάνω υλοποίησης υπάρχει στο project "HPY411_Lab_9.c" του παραδοτέου φάκελου.

Αποτελέσματα Προσομοίωσης Υλοποίησης με Polling:

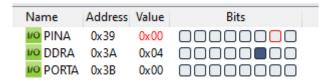
Για την προσομοίωση του προγράμματος αρχικά δοκιμάστηκε το σενάριο όπου η έξοδος μεταβαίνει από το '0' στο '1'. Αρχικά, εκτελέστηκε το αρχείο "lab9_init.stim" το οποίο αρχικοποιεί τον διακόπτη στην κατάσταση "OFF" (ακροδέκτης A(PA0) στο '0', ακροδέκτης A'(PA1) στο '1'). Μετά την εκτέλεση του αρχείου, παρατηρούμε ότι ο ακροδέκτης A(PA0) έχει την τιμή '0' και ο ακροδέκτης A'(PA1) έχει την τιμή '1':

Name	Address	Value	Bits
MO DINA	0x39	0x02	
₩ DDRA	0x3A	0x04	
WO PORTA	0x3B	0x00	0000000

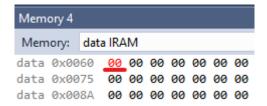
Επιπλέον, ο διακόπτης είναι στην κατάσταση "STATE_SET", όπως φαίνεται από την τιμή στην μνήμη:



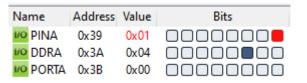
Στην συνέχεια, εκτελούμε το αρχείο "lab9_on.stim" το οποίο προσομοιώνει την αναπήδηση "0->1->0->1" στον ακροδέκτη Α(PA0) με την τιμή του Α'(PA1) να είναι '0'. Αρχικά παρατηρούμε ότι οι ακροδέκτες Α και Α΄ έχουν την τιμή '0'.



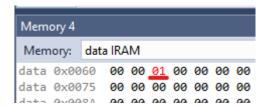
Επιστρέφοντας από το interrupt, παρατηρούμε ότι ο διακόπτης βρίσκεται στην κατάσταση "STATE_BOUNCING" καθώς η τιμή της διεύθυνσης "0x0060" είναι '0'.



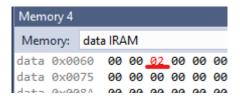
Στην συνέχεια παρατηρούμε ότι ο ακροδέκτης Α έχει την τιμή '1' και ο Α΄ έχει την τιμή '0'.



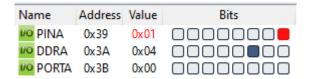
Και η κατάσταση παραμένει να είναι "STATE_BOUNCING". Επιπλέον, παρατηρούμε ότι εφόσον είμαστε στην κατάσταση "STATE_BOUNCING" έχει ξεκινήσει να μετριέται το χρονικό πλαίσιο των 10 msec. Όπως παρατηρούμε από την τιμή της διεύθυνσης "0x0062".



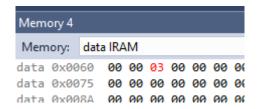
Στην συνέχεια ότι οι ακροδέκτες Α και Α΄ έχουν ξανά την τιμή '0', η κατάσταση συνεχίζει να είναι "STATE_BOUNCING" και ο μετρητής για τον το χρονικό πλαίσιο των 10 msec αυξάνεται.



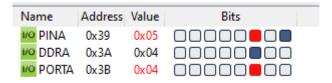
Τέλος, ο ακροδέκτης Α έχει την τιμή '1' και ο Α' έχει την τιμή '0'.



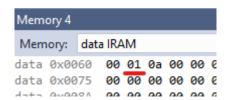
Και η κατάσταση παραμένει "STATE_BOUNCING".



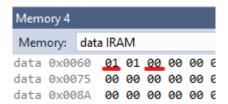
Συνεχίζοντας την εκτέλεση παρατηρούμε ότι η τιμή των ακροδεκτών Α και Α΄ παραμένει η ίδια. Όταν το χρονικό πλαίσιο των 10 msec τελειώσει, παρατηρούμε ότι η τιμή του ακροδέκτη Α βγαίνει στην έξοδο PA2.



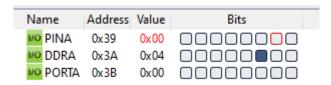
Και η τιμή της διεύθυνσης "0x0061" παίρνει την νέα σταθερή τιμή.



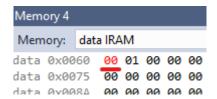
Επιστρέφοντας από το interrupt, παρατηρούμε ότι η κατάσταση του διακόπτη αλλάζει σε "STATE_SET" και η τιμή του μετρητή για το πλαίσιο των 10 msec γίνεται '0'.



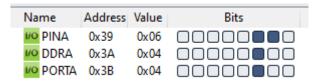
Στην συνέχεια για την προσομοίωση της μετάβασης του ακροδέκτη Α από το '1' στο '0' (δηλαδή από την κατάσταση "ON" στην κατάσταση "OFF"), εκτελούμε το αρχείο "lab9_off.stim", το οποίο προσομοιώνει την αναπήδηση "1->0->1->0" στον ακροδέκτη Α'(PA1) με την τιμή του Α(PA0) να είναι '0'. Αρχικά παρατηρούμε ότι οι ακροδέκτες Α και Α΄ έχουν την τιμή '0'.



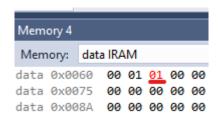
Επιστρέφοντας από το interrupt παρατηρούμε ότι ο διακόπτης μεταβαίνει από την κατάσταση "STATE_SET" στην κατάσταση "STATE_BOUNCING".



Συνεχίζοντας την εκτέλεση, παρατηρούμε ότι ο ακροδέκτης Α έχει την τιμή '0' και ο Α΄ έχει την τιμή '1'.



Και ο μετρητής αρχίζει να μετράει για τον χρόνο των 10 msec.



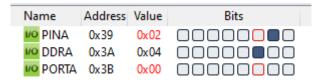
Αντίστοιχα με το προηγούμενο σενάριο, συνεχίζοντας την εκτέλεση παρατηρούμε ότι στο τέλος των αναπηδήσεων ο ακροδέκτης Α έχει την τιμή '0' ενώ ο Α' την τιμή '1'.



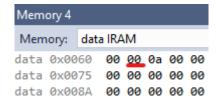
Συνεχίζοντας την εκτέλεση μέχρι το τέλος των 10 msec παρατηρούμε ότι η τιμή των ακροδεκτών έχει παραμείνει σταθερή.



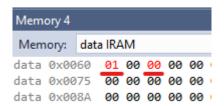
Μετά τα 10 msec, παρατηρούμε ότι η τιμή της εξόδου παίρνει την τιμή του ακροδέκτη Α (δηλαδή την τιμή '0').



Και η νέα σταθερή τιμή αποθηκεύεται στην διεύθυνση "0x0061" της μνήμης.



Τέλος, μετά την επιστροφή από το interrupt στο κύριο πρόγραμμα, παρατηρούμε ότι η κατάσταση του διακόπτη έχει μεταβεί στην κατάσταση "STATE_SET" και η τιμή του μετρητή είναι '0'.

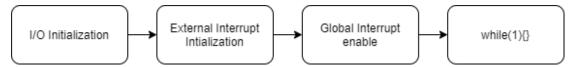


ΣΗΜΕΊΩΣΗ: Τα αρχεία ".stim" που χρησιμοποιήθηκαν παραπάνω, υπάρχουν στο παραδοτέο φάκελο.

Υλοποίηση Λύσης με Εξωτερικά Interrupt:

Για την υλοποίηση με τα εξωτερικά interrupts, αρχικά δημιουργήθηκε η συνάρτηση "void ExternalInterrupt_init(void)", στην οποία αρχικοποιείται η τιμή των bits ISC00, ISC01, ISC10 και ISC11 στην τιμή '1', το οποίο ορίζει ότι τα interrupts θα καλούνται στην θετική ακμή της αλλαγής της τιμής τους. Επιπλέον, τα bits INT0 και INT1 του καταχωρητή GICR θέτονται στην τιμή '1' για την ενεργοποίηση των εξωτερικών interrupt.

Το σώμα του κύριου προγράμματος(main()) περιγράφεται από το παρακάτω διάγραμμα:



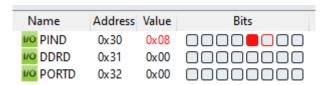
Για τον ακροδέκτη Α χρησιμοποιείται το εξωτερικό interrupt INTO, το οποίο αντιστοιχεί στον ακροδέκτη PD2 του PORTD, ενώ για τον ακροδέκτη A' χρησιμοποιείται το εξωτερικό interrupt INT1, το οποίο αντιστοιχεί στον ακροδέκτη PD3 του PORTD. Για την έξοδο του διακόπτη, χρησιμοποιείται ο ακροδέκτης PA2 του PORTA. Σε κάθε περίπτωση, η έξοδος δίνεται από τη κατασταλαγμένη τιμή του ακροδέκτη A, δηλαδή από την τιμή του pin PD2. Και τα δύο interrupts ενεργοποιούνται στην θετική ακμή της αλλαγής της τιμής των ακροδεκτών PD2 και PD3, δηλαδή στην μετάβαση από την τιμή '0' στην τιμή '1'. Αυτό γίνεται γιατί, με βάση την λειτουργία του διακόπτη, παρατηρήθηκε ότι ο ακροδέκτης που κάνει αναπήδηση είναι ο ακροδέκτης ο οποίος μεταβαίνει από την τιμή '0' στην τιμή '1'. Επομένως, όταν ενεργοποιείται το INTO που αντιστοιχεί στην αναπήδηση το ακροδέκτη A, η έξοδος στο PA2 θα πάρει την τιμή '1'. Αντίστοιχα, όταν ενεργοποιείται το INT1(δηλαδή ο A' κάνει αναπήδηση) η τιμή της εξόδου θα είναι '0'.

ΣΗΜΕΊΩΣΗ: Ο κώδικας της παραπάνω υλοποίησης υπάρχει στο project "HPY411_Lab_9_b.c" του παραδοτέου φάκελου.

Αποτελέσματα Προσομοίωσης Υλοποίησης με Εξωτερικά Interrupt:

Αρχικά, εκτελέστηκε το αρχείο "lab9_b_on.stim" στο οποίο αρχικά δίνουμε την τιμή '0' στον ακροδέκτη Α΄ και την τιμή '1' στον ακροδέκτη Α΄ για μερικές χιλιάδες κύκλους για να

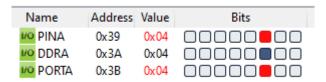
αρχικοποιήσουμε τον διακόπτη στο "OFF", όπως φαίνεται και στην παρακάτω εικόνα. Το πρόγραμμα μεταβαίνει στην ρουτίνα εξυπηρέτησης του interrupt INT1, όπως θα περιμέναμε και θέτει την τιμή της εξόδου '0'. Στην ουσία δεν έχουμε καμία αλλαγή στην έξοδο του προγράμματος και ο διακόπτης αρχικοποιείται στο "OFF".



Στην συνέχεια, δίνουμε και στους δύο ακροδέκτες την τιμή '0'. Επειδή τα interrupts ενεργοποιούνται μόνο στην μετάβαση από '0' σε '1', το πρόγραμμα θα αγνοήσει αυτή την αλλαγή και θα συνεχίσει την κανονική ροή του. Έπειτα, ο ακροδέκτης Α παίρνει την τιμή '1' και ο ακροδέκτης Α' την τιμή '0' όπως φαίνεται και παρακάτω.

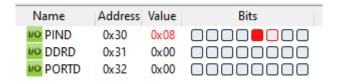
Nar	me	Address	Value	Bits
I/O P	IND	0x30	0x04	
NO D	DRD	0x31	0x00	00000000
I/O P	ORTD	0x32	0x00	00000000

Συνεπώς, παρατηρούμε ότι το πρόγραμμα μεταβαίνει στην ρουτίνα εξυπηρέτησης του interrupt INTO και βγάζει στην έξοδο την τιμή '1'.



Στην συνέχεια, μέχρι να τελειώσει η εκτέλεση του αρχείου ".stim", παρατηρούμε ότι η τιμή '1' παραμένει στην έξοδο(PA2) χωρίς να επηρεάζεται από την αναπήδηση του ακροδέκτη Α.

Μετά από μερικούς χιλιάδες κύκλους, εκτελούμε το αρχείο "lab9_b_off.stim". Παρατηρούμε ότι αρχικά το αρχείο ".stim" δίνει την τιμή '1' στον ακροδέκτη Α και την τιμή '0' στον ακροδέκτη Α'(δηλαδή τις προηγούμενες τιμές τους) και επομένως δεν έχουμε καμία αλλαγή στην έξοδο PA2. Στην συνέχεια και οι δύο ακροδέκτες παίρνουν την τιμή '0' και παρατηρούμε ότι δεν υπάρχει αλλαγή στο πρόγραμμα μας. Αυτό συμβαίνει καθώς δεν είχαμε κάποια μετάβαση από '0' σε '1' επομένως δεν ενεργοποιήθηκε κανένα interrupt. Συνεχίζοντας με την εκτέλεση του αρχείου ".stim", παρατηρούμε ότι ο ακροδέκτης Α παίρνει την τιμή '0' ενώ ο ακροδέκτης Α' παίρνει την τιμή '1'.



Στην συνέχεια, ενεργοποιείται το interrupt INT1 καθώς ο ακροδέκτης Α΄ άλλαξε από την τιμή 'Ο' στην τιμή '1' και η έξοδος PA2 γίνεται 'Ο'.

Name	Address	Value	Bits
I/O PINA	0x39	0x00	0000000
₩ DDR	A 0x3A	0x04	
WO PORT	A 0x3B	0x00	00000000

Στην συνέχεια, μέχρι να τελειώσει η εκτέλεση του αρχείου ".stim", παρατηρούμε ότι η τιμή '0' παραμένει στην έξοδο(PA2) χωρίς να επηρεάζεται από την αναπήδηση του ακροδέκτη Α'.

ΣΗΜΕΊΩΣΗ: Τα αρχεία ".stim" που χρησιμοποιήθηκαν παραπάνω, υπάρχουν στο παραδοτέο φάκελο.

Συμπεράσματα:

Συγκρίνοντας τις δύο υλοποιήσεις παρατηρούμε ότι στην υλοποίηση με τα εξωτερικά interrupts η τιμή της εισόδου χρειάζεται πολύ λιγότερους κύκλους για να κατασταλάξει σε σχέση με την υλοποίηση με polling, καθώς δεν χρειάζεται μετρήσουμε τον χρόνο των 10 msec για να εμφανίσουμε την έξοδο. Επιπλέων, η υλοποίηση με interrupt δεν χρησιμοποιεί καθόλου χώρο από την μνήμη RAM, σε αντίθεση με την υλοποίηση του polling που χρησιμοποιεί 3 bytes της μνήμης. Παρόλα αυτά, για την υλοποίηση με interrupts χρησιμοποιούμε και τα δύο διαθέσιμα εξωτερικά interrupt.