

ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΗΓΜΕΝΩΝ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

3Η ΑΣΚΗΣΗ

05/05/2022

ΓΕΩΡΓΙΟΣ ΚΟΝΤΟΓΙΑΝΝΗΣ - 1070908

ΧΑΡΑΛΑΜΠΟΣ ΠΑΠΠΑΣ - 1053359







* Βασική Ροή προγράμματος

1. Κατά την έναρξη του προγράμματος, δηλώνουμε τις μεταβλητές stop\_adc=1, first\_time=0, pulses=0. Οι δύο (2) πρώτες χρησιμοποιούνται για τον έλεγχο και τερματισμό του ADC ενώ η τρίτη (3) μεταβλητή για τον ελέγχο συμμετρίας των κυματομορφών. Επίσης τίθονται ώς έξοδοι τα PIN0, PIN1, PIN2 των οποίων τα αντίστοιχα LED είναι όλα σβηστά. Τα switch5, switch6 του PORTF **ενεργοποιούνται τα interrupts.**
2. Το πρόγραμμα καλέι τη συνάρτηση ADC\_init(void), η οποία αρχικοποιεί τον ADC σε Free-run mode με τα κατάλληλα σήματα, θέτει το κατώφλι WINLT=5 σηματοδωτόντας έτσι ότι αν ο αισθητήτας ανιχνεύσει απόσταση μικρότερη ή ίση απο αυτή θα πυροδοτηθεί το ISR(ADC0\_WCOMP\_vect). Στη συνέχεια ελέγχει την απόσταση και δρά αναλόγως. Στο σημείο αυτό πρέπει να σημειωθεί πώς κατά την έναρξη του προγράμματος ο RES του ADC είναι μηδέν (0), για το λόγο αυτό του δίνουμε μια μεγαλύτερη τιμή.
3. Στη συνέχεια, καλούνται οι TCB0\_init(void), TCB1\_init(void) οι οποίες αρχικοποιούν τους δύο counters με την *ίδια τιμή περιόδου και duty cycle = 50%.* Τέλος, ενεργοποιούν την μέτρηση τους και η κλήση επιστρέφει στη main().
4. Επιστρέφοντας στη main(), ενεργοποιούμε τα interrupts και εαν καμία άλλη ενέργεια δε πάρει μέρος απο το χειριστή, κατά τη λήξη των counters (λήγουν μαζί καθώς έχουν την ίδια περίοδο) πυροδοτείται η ISR(TCB0\_INT\_vect).
5. Εκτελόντας την ISR, αρχικά ελέγχει αν πρόκειται για άρτιο ή περιττό αριθμό παλμού. Στην άρτια περίπτωση ανάβουμε τα LED, ενώ στην περριτή τα σβήνουμε. Έτσι, αρχικά κάνουμε αυτόν τον έλεγχο και αν είναι άρτιος ανάβουμε μόνο το LED0 που αντιστοιχεί στον παλμό του TCB0, ενώ για να ανάψουμε και το LED1 (που αντιστοιχεί στον παλμό του TCB1) ελέγχουμε τα flags του για να βεβαιωθούμε οτι και αυτός «έσκασε». Αυτό είναι απαραίτητο διότι χρησιμοποιούμε την ISR του TCB0. Έτσι, αν τo INTFLAGS του TCB1 είναι ένα (1) τότε ανάβουμε και το LED1. Με αντίστοιχη λογική λειτουργεί και στη περριτή περίπτωση. Τέλος, αυξάνουμε τον αριθμό των παλμών κατά ένα (1) και το πρόγραμμα συνεχίζει στη main(). Μετά τη λήξη των μετρητών, ξανα ενεργοποιείται η ISR(TCB0\_INT\_vect) και η διαδιακασία αυτή επαναλαμβάνεται μέχρις ώτου ο ADC να ανιχνεύσει εμπόδιο.

* Εναλλακτική Ροή (ADC Interrupt και απόφαση κατεύθυνσης)

1. Όταν ο καταχωρητής RES του ADC λάβει τιμή μικρότερη του πέντε (5), πυροδοτείται η ISR(ADC0\_WCOMP\_vect). Κατά τη κλήση της, ελέγχει τα flags, ανάβει το LED2 το οποίο σηματοδοτεί εμπόδιο και αναμονή για απόφαση εντολής.
2. Ο χρήστης «πατάει» το switch της επιλογής του και θέτει τον RES μια τιμή μεγαλύτερη του πέντε (5) και στους επόμενους κύκλους ρολογιού πυροδοτείται η ISR(PORTF\_PORT\_vect).
3. Κατά την εκτέλεση της, αποθηκεύει στην intflags μεταβλητή τα flags του PORTF και με bit masking ελέγχει πιο απο τα δύο switches πατήθηκε. Αν πατήθηκε το switch5, δηλαδή ο χρήστης επιθυμεί η συσκευή να στρίψει δεξιά, τότε εκτελέιται το πρώτο if και διπλασιάσει τη περίοδο της κυματομορφής του TCB1 (του αριστερά τροχού) και η άλλη κρατάει τη τιμή που της καταχωρήθηκε στην έναρξη του προγράμματος. Στην περίπτωση που έχει πατήσει το switch6, δηλαδή αριστερή στροφή, τότε διπλασιάζεται η περίοδος της κυματομορφής του TCB0 (δεξιά τροχού). Τίθεται το stop\_adc = 0 , σβήνει το LED2 (απόφασης) και το πρόγραμμα συνεχίζεται στη main.
4. Στη main πυροδοτείται η ISR(TCB0\_INT\_vect) η οποία θα αναβοσβήνει τα LED0, LED1 σεβόμενη τις περιόδους των κυματομορφών.
5. Ξε-πατώντας το switch που πατήθηκε προηγουμένως και θέτοντας χειροκίνητα τη τιμή της περιόδου που αλλάχτηκε να είναι ίση με την αρχική, το πρόγραμμα ουσιαστικά επιστρέφει στη Βασική Ροή Προγράμματος περιμένοντας την πυροδότηση του επόμενου interrupt.