

ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΗΓΜΕΝΩΝ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

4Η ΑΣΚΗΣΗ

19/05/2022

ΓΕΩΡΓΙΟΣ ΚΟΝΤΟΓΙΑΝΝΗΣ - 1070908

ΧΑΡΑΛΑΜΠΟΣ ΠΑΠΠΑΣ - 1053359









* **Σημαντικές Μεταβλητές**
* **pass\_counter** 🡪 Αρχικά είναι «0» και είναι υπεύθυνη για τη σωστή σειρά εισαγωγής των ψηφίων. Οταν γίνει «4» σηματοδοτεί οτι ολοκληρώθηκε η εισαγωγή του κωδικού.
* ***pass\_input*** 🡪 Αρχικά είναι «0» και γίνεται «1» όταν εισαχθεί ο κωδικός σωστά.
* ***timer\_pass*** *🡪* Αρχικά είναι «0» και γίνεται «1» όταν λήξει το χρονικό περιθώριο για να φύγουν οι ιδιωκτήτες.
* ***is\_robbed*** *🡪* Αρχικά είναι μηδέν και μόλις ανιχνευθεί κοντινή κίνηση, δηλαδή τιμή κάτω απο το threshold του ADC WINLT γίνεται «1».
* ***is\_robbed\_timer*** *🡪* Αρχικά είναι «0» και γίνεται «1» όταν λήξει το χρονικό περιθώριο για να εισαχθεί ο σωστός κωδικός πριν την έναρξη της σειρήνας.
* ***pulses*** 🡪 Χρησιμοποιείται για την ανίχνευση παλμού στον PWM ελέγχοντας τη συμμετρία της τιμής της.
* **Λετουργία αναμονής εισαγωγής σωστού συνδυασμού και ενεργοποίηση - απενεργοποίηση του χρονιστή**

Για τις λειτουργίες αυτές υπεύθυνη είναι η συνάρτηση enable\_alarm(). Η συνάρτηση enable\_alarm() είναι η πρώτη που καλέιται απο τη main αφότου μόλις ενεργοποιήσει τα switch interrupts για τα SW5, SW6. Κατά την είσοδο ενεργοποιούνται τα ISR Interrupts και «κλειδώνει» εκεί μέχρις ώτου να πατηθεί ένας από τους παραπάνω διακόπτες. Κατα το πάτημα, καλείται η ISR(PORTF\_PORT\_vect) όπου μία απο τις βασικές τις λειτουργίες είναι ο έλεγχος σωστής εισαγωγής κωδικού.

* + **Λογική εισαγωγής κωδικού:**

Κάθε φορά που καλείται ηISR(PORTF\_PORT\_vect), με masking ελέγχεται ποιο switch πατήθηκε. Γνωρίζοντας λοιπόν τη σωστή ακολουθία του κωδικού (sw6-sw5-sw5-sw6) και ότι το πάτημα του sw6 αντιστοιχεί στην 0η ή 3η κλήση της ISR ενώ το sw5 στή 1η ή 2η, με μία δομή if() else if() εύκολα κάνουμε αυτό τον έλεγχο. Εκτελόντας τη κατάλληλη συνθήκη σε κάθε κάλεσμα της ISR, αυξάνουμε τον pass\_counter κατά «1». Κατά την έξοδο από την ISR «καθαρίζουμε» τα PORTF.FLAGS για την εισαγωγή του επόμενου ψηφίου. Όταν ο pass\_counter γίνει ίσος με «4», σηματοδοτεί οτι όλα τα ψηφία μπήκαν με τη σωστή σειρά και το pass\_input γίνεται «1» ξεκλειδώνοντας έτσι τη συνάρτηση enable\_alarm() από την αναμονή εισαγωγής κωδικού. Αν κατά την εισαγωγή ψηφίων η σωστή ακολουθία «χαλάσει», o pass\_counter μηδενίζεται και η λογική αυτή επαναλαμβάνεται.

* + **Λογική εν/απ-ενεργοποίσης χρονιστή αποχώρησης**

Μετά τη σωστή εισαγωγή του κωδικού, η enable() αρχικοποιεί τον TCA0 timer και ορίζει μια χρονική ποσότητα για την έξοδο των χειριστών απο τον χώρο. Κατά τη λήξη του, καλείται η ISR(TCA0\_CMP0\_vect) και ελέγχει άν η πυροδότηση της προήλθε από την αναμονή αποχώρησης (if pass\_input==1) (κάνουμε αυτό το διαχωρισμό γιατί στη συνέχεια ο TCA0 χρησιμοποιείται και για το περιθώριου χρόνου εισαγωγής του κωδικού πρίν την ενεργοποίηση της σειρήνας). Κατά την εκτέλεση της, καθαρίζουμε και απενεργοποιούμε τον TCA0, επαναφέρουμε την μεταβλητή pass\_input=0, δηλώνοντας οτι η επόμενη λειτουργία απαιτεί ξανά την εισαγωγή κωδικού και τέλος κάνουμε την timer\_after\_pass = 1 δηλώνοντας οτι το χρονικό όριο αποχώρησης τελείωσε. Το ροή πρόγραμματος επιστρέφει στην enable\_alarm() και αμέσως μετά στη main().

* **Ενεργοποίηση ADC, LED0, Timer και έλεγχος εισαγωγής συνδυασμού στο χρονικό περιθώριο**
  + **Λογική ΑDC και LED0**

Κατά την επιστροφή του προγράμματος στη main, καλείται η συνάρτηση ADC\_init(). Εκτελόντας την ADC\_init(), αρχικοποιείται ο ADC και θέτει WINLT=10. Στη συνέχεια ενεργοποιεί τα ISR Interrupts, ξεκινάει τη μετατροπή και όσο το WINLT >10 “εξαιτίας” της is\_robbed περιμένει σε αυτή τη κατάσταση. Κατά την ανίχνευση και μετατροπή τιμής εισόδου <=10, πυροδοτείται η ISR(ADC0\_WCOMP\_vect) Κατα την εκτέλεση της, καθαρίζει τα flags του ADC, θέτει το is\_robbed=1 για να συνεχίσει το επόμενο απαραίτητο βήμα και ανάβει το LED0. Η ροή προγράμματος επιστρέφει στην ADC\_init() και ύστερα στη main().

* + **Λογική ενεργοποιησης Timer πρίν τη σειρήνα και εισαγωγή 3ων προσπαθειών**

Επιστρέφοντας στη main(), καλείται η συνάρτηση disable\_alarm(). Κατά την εκτέλεση της, αρχικοποιείται και ενεργοποιείται ο TCA0 με χρονικό περιθώριο 10 μονάδων για τη σωστή εισαγωγή του κωδικού. Το πρόγραμμα περιμένει σε αυτό το σημείο με τη βοήθεια της μεταβλητής is\_robbed\_timer μέχρις ώτου να εισαχθεί ο σωστός κωδικός ή να λήξει ο χρόνος και να πυροδοτηθεί η ISR(TCA0\_CMP0\_vect). Πατώντας οποιοδήποτε απο τα δύο απαραίτητα switches, καλείται η ISR(PORTF\_PORT\_vect) για την εισαγωγή κωδικού με ίδια λογική κατα την ενεργοποίση του συναγερμού. Στη περίπτωση όπου εισαχθεί σωστά ο κωδικός με λιγότερες απο τρείς προσπάθειες, η μεταβλητή pass\_input == 1 και κατά τη πυροδότηση της ISR(TCA0\_CMP0\_vect ) μετά τη λήξη του χρόνου, θα ισχύει η 1η συνθήκη. Στη περίπτωση αυτή καθαρίζει τα flags, απενεργοποιέι τον timer και θέτει τη pass\_input=0 ώστε να ζητηθεί ξανά κωδικός στην επομένη ενεργοποίσηση του συναγερμού. Η ροή προγράμματος συνεχίζεται στη main() καλόντας ξανά την enable\_alarm() όπου θα επαναληφθούν ακριβώς όλα τα βήματα που περιγράφηκαν μέχρι αυτό το σημείο.

Εναλλακτικά, στη περίπτωση που «σκάσει» ο timer (χωρις κωδικό), καλέιται η ISR(TCA0\_CMP0\_vect ) και αυτή τη φορά ισχύει η δεύτερη συνθήκη. Δηλαδή pass\_input == 0 το οποίο σηματοδοτεί πώς δεν έχει εισαχθεί σωστός κωδικός. Στο σημέιο αυτό καλείται η siren() συνάρτηση. Τέλος, κάθε φορά που «χαλάει» η σειρά εισαγωγής των ψηφιίων, η μεταβλητή tries αυξάνεται κατά «1» και ο pass\_counter μηδενίζεται για την επαναφορά διαδικασίας εισαγωγής κωδικού. Στη περίπτωση όπου tries == 3, καλείται η συνάρτηση siren().

* PWM σειρήνα και LED0

Κατά την εκτέλεση της συνάρτησης siren(), με τη βοήθεια του TCB0, τον θέτουμε σε PWM-mode με PER=0x10 και T=0x20 και τον ενεργοποιούμε.

* + **Λογική Σειρήνας με LED**

Μετά την ενεργοποίσηση της σειρήνας, δηλαδή τη παραπάνω αρχικοποίηση του PWM, σε κάθε ολοκλήρωση παλμού του, καλείται η ISR(TCB0\_INT\_vect). Κατά την εκτέλεση της, αρχικά ελέγχεται αν έχει εισαχθεί σωστός κωδικός με τη βοήθεια της συνθήκης pass\_count != 4. Όπως αναφέρθηκε στο 1ο ερώτημα, pass\_counter==4 σηματοδοτεί σωστή εισαγωγή κωδικού. Άρα, όσο ισχύει αυτό (δηλαδή όχι σωστός κωδικός), ελέγχοντας τη συμμετρία της τιμής της μεταβλητής pulses, ανάβει και σβήνει αντίστοιχα το LED0 και αυξάνει το pulses κατά «1» σε κάθε κλήση της ISR.

Στη περίπτωση που εισαχθέι σωστά ο κωδικός με το τρόπο που εξηγήθηκε παραπάνω, απενεργοποιείται ο TCB καθώς πλέον pass\_counter==4 (σταματάει δηλαδή η σειρήνα) και μέσω της ISR(PORTF\_PORT\_vect), όταν εισήχθη σωστά ο κωδικός, το is\_robbed\_timer ισούται με «1» σηματοδοτώνας ότι απενεργοποιήθηκε ο συναγερμός. Η ροή προγράμματος επιστρέφει στη main() και ξανά την enable\_alarm() όπου θα επαναληφθούν ακριβώς όλα τα βήματα που περιγράφηκαν μέχρι αυτό το σημείο.