

Συστήματα Μικροϋπολογιστών

Αναφορά για 2ο εργαστήριο

Ιωάννης-Παναγιώτης Μπουντουρίδης 8872 - Κωνσταντίνος Χατζηπαντωνίου 8941

16/12/2018

Εξήγηση Κώδικα

A) Προετοιμασία για εισαγωγή κωδικού και αποθήκευση

Για να εισάγουμε τον κωδικό, πατάμε ταυτόχρονα κάποια από τα πλήκτρα SW1-SW4. Όσο ένα πλήκτρο είναι πατημένο, φορτώνεται στον καταχωρητή που του αντιστοιχεί, ένας αριθμός. Ο αριθμός αυτός έχει παντού μηδέν εκτός από τη θέση π αντιστοιχεί στο πλήκτρο (SW1-> 0b00000001 ... SW3-> 0b00000100). Αν αφήσουμε το πλήκτρο, ο καταχωρητής του γίνεται 0, οπότε δεν υπάρχει πρόβλημα αν πατήσουμε ένα πλήκτρο καταλάθος. Όταν είμαστε έτοιμοι να τον αποθηκεύσουμε, πατάμε το SW7 (ενώ κρατάμε πατημένα τα άλλα πλήκτρα) και μεταφερόμαστε στο label SAVE_PASSWD. Εκεί προσθέτουμε τους 4 καταχωρητές που σώσαμε αν πατήθηκε κάθε πλήκτρο και ελέγχουμε αν είναι μεγαλύτερο του 0(αν εισάγαμε όντως κάποιον κωδικό).Αν δεν βάλαμε κωδικό, αναβοσβήνουμε τα leds για να δείξουμε ότι υπήρξε πρόβλημα και γυρνάμε στο προηγούμενο loop. Αν εισάγαμε σωστά κάποιον κωδικό, γυρνάμε πάλι στο πρώτο loop, όπου μπορούμε να εισάγουμε νέο κωδικό ή να πάμε στην κυρίως λειτουργία πατώντας το SW0.

B) Κανονική Λειτουργία

Αφού πατήσουμε το SW0 και εφόσον έχουμε βάλει κάποιο κωδικό, πηγαίνουμε στο label INIT2, όπου αρχικοποιούμε τους μετρητές χρόνου και αντιγράφουμε τον κωδικό σε έναν ακόμα καταχωρητή. Μετά πηγαίνουμε στο loop κάτω από το label NORMAL_OP όπου κάνουμε τα εξής:

1. Καλούμε την υπορουτίνα TIMER25us που καθυστερεί για 25 us.
2. Αυξάνουμε τον μετρητή για τα 5 δευτερόλεπτα(time_counter5) και άμα χρειάζεται τον μετρητή για το 1 δευτερόλεπτο(time_counter1).
3. Ελέγχουμε αν πέρασαν 5 δευτερόλεπτα (time_counter5 > 200).Αν ναι, πηγαίνουμε στο label TIMEOUT
4. Ελέγχουμε αν πατήθηκε κάποιο κουμπί, και το αποθηκεύουμε στον καταχωρητή arg
5. Αν ο καταχωρητής arg είναι διάφορος του μηδενός (δλδ πατήσαμε κάποιο κουμπί), πηγαίνουμε στο label BTN_INPUT.
6. Καλούμε την υπορουτίνα LED_FUN
7. Ελέγχουμε αν έχει εισαχθεί όλος ο κωδικός. Αν ναι πηγαίνουμε στο label INIT3

Χρήση Καταχωρητών

- passwd_temp -> Αποθηκεύουμε τον κωδικό που θα ελέγξουμε.
- arg -> Αποθηκεύουμε ποιο κουμπί πατήσαμε.
- led_function -> Αποθηκεύουμε αν, και πόσα, led πρέπει να είναι ανοιχτά.
- t1cntr_flag -> μηδενίζει τον time_counter1 αν δεν τον χρειαζόμαστε.

Υπορουτίνα TIMEOUT και WRONG_BTN

Η υπορουτίνα καλείται όταν περάσουν 5 δευτερόλεπτα ή πατήσουμε λάθος πλήκτρο. Αρχικά μηδενίζουμε τον `time_counter5`. Αυξάνουμε τον καταχωρητή `led_function`, αφού έγινε λάθος. Κάνουμε reset τον καταχωρητή `passwd_temp`. Τέλος γυρνάμε ξανα στο loop `NORMAL_OP`.

Υπορουτίνα BTN_INPUT

Η υπορουτίνα καλείται όταν έχει πατηθεί ένα πλήκτρο. Η τιμή του πλήκτρου είναι αποθηκευμένη στον καταχωρητή `args`. Περιμένουμε να αφήσουμε το πλήκτρο και πηγαίνουμε στο label `CHECK_IF_CORRECT`

Υπορουτίνα CHECK_IF_CORRECT

Με την εντολή `and` και μια σύγκριση ελέγχουμε αν το κουμπί που πατήσαμε υπάρχει στον κωδικό. Αν δεν υπάρχει πηγαίνουμε στο label `WRONG_BTN`. Αν υπάρχει το αφαιρούμε από τον καταχωρητή `passwd_temp` και ελέγχουμε αν είναι μεγαλύτερο από τον `args` για να διαπιστώσουμε τα πατήσαμε στη σωστή σειρά. Αν το πατήσαμε στη λάθος σειρά πηγαίνουμε στο label `WRONG_BTN`. Αν όλα πήγαν σωστά γυρνάμε στο loop `NORMAL_OP`.

Υπορουτίνα LED_FUN

Εδώ ελέγχουμε αν ο καταχωρητής `led_fun` είναι 0->κανένα λάθος 1-> ένα λάθος 2+>δύο ή περισσότερα λάθη. Αν είναι μηδέν, απλά γυρνάμε στο loop `NORMAL_OP`. Αν είναι 1 ελέγχουμε αν ο `time_counter1` είναι κάτω από 40, οπότε σβήνουμε τα led, αν είναι μεγαλύτερο από το 40, τον κάνουμε 215(ώστε από το overflow να γυρίσει στο 0) και ανάβουμε το led.

Αν είναι 2+, κάνουμε ότι κάναμε πριν, απλά ανάβουμε όλα τα leds.

Γ) Λειτουργία Αισθητήρων

Εφόσον έχουμε εισάγει τον κωδικό σωστά, ο καταχωρητής `passwd_temp` είναι μηδέν, αφού αφαιρούμε κάθε φορά το σωστό ψηφίο. Στο label `INIT_3` αρχικοποιούμε τα `temp3,temp4 = 0xFF` που θα χρησιμοποιήσουμε ως flags και ανάβουμε τα κατάλληλα leds. Στην αρχή του loop `SENSOR_MODE` ελέγχουμε αν πατήθηκε κάποιο κουμπί. Αν πατήθηκε το SW0 πηγαίνουμε στο `INIT3`. Αν πατήθηκε κάποιο από τα SW4-7 βάζουμε τον κατάλληλο αριθμό στον `temp1`, πατήθηκε το SW2, στον `temp2`. Αν πατήθηκε κάποιο από τα bypass, κάνουμε τα αντίστοιχα flags 0x00. Αν τα `temp1 temp2` είναι μεγαλύτερο του 0 σημαίνει ενεργοποιήσαμε κάποιον αισθητήρα και πηγαίνουμε σε ένα από τα labels `ERROR1 ERROR2`.

Λειτουργία των flags

Τα flags λειτουργούν σαν mask πάνω στα `temp1, temp2` με `and`. Αν δεν πατηθούν μένουν τα bypass, τα flags μένουν 0xFF και αφίνουν τα `temp1 temp2` ως έχουν. Αν πατηθούν γίνονται 0x00 και κάνουν τα `temp 0` και δεν έχει σημασία αν πατηθούν τα πλήκτρα-αισθητήρες

Υπορουτίνα ERROR1,2

Ελέγχουμε αν αφήσαμε το πλήκτρο-αισθητήρα που πατήσαμε. Μετά ελέγχουμε αν κάναμε πρώτη φορά το λάθος, και προσθέτουμε το κουμπί που πατήσαμε στο `led_fun`. Μετά κάνουμε `ori 0x04`, αφού το led 2 είναι πάντα ανοιχτό σε κάθε παραβίαση.

Δυσκολίες/Σχόλια/Περαιτέρω Βελτίωση

Δεν αντιμετωπίσαμε κάποια δυσκολία και το πρόγραμμα έτρεξε σωστά στον AVR, όπως υπολογίζαμε από την προσομοίωση. Το πιο ενδιαφέρον ήταν να αναβοσβήνουμε τα leds και να μετράμε χρόνο ενώ ταυτόχρονα να ελέγχουμε για είσοδο. Βάζοντας μια μικρή καθυστέρηση και μετρώντας πόσες φορές έτρεξε, καταφέραμε να μην μπλοκάρει το πρόγραμμα εξαιτίας του delay loop. Δε φαίνεται κάποιο κομμάτι του προγράμματος να υστερεί και να χρειάζεται βελτίωση από άποψη χρηστικότητας