

# ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

## ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2

GROUP 5 - ΟΜΑΔΑ 6:  
Θωμάς Πλιάκης (9018)

### Λειτουργία προγράμματος:

#### I)Εισαγωγή κωδικού που θα δέχεται το σύστημα:

Το πρόγραμμα ξεκινάει με την μετονομασία 3 καταχωρητών που χρησιμοποιούνται για αναβοσβήνουν τα LEDS, για την αποθήκευση του σωστού κωδικού του συστήματος αλλά και του κωδικού που εισάγεται στο σύστημα ασφαλείας. Μετά γίνεται η αρχικοποίηση της στοίβας στην SRAM και της της PORTB ως θύρας εξόδου για να μπορούμε να ανάψουμε αργότερα τα leds. Έπειτα το πρόγραμμα περιμένει μέχρι να πατηθεί ο SW1 ώστε να εισαχθεί ο κωδικός που θα δέχεται ως σωστό το σύστημα ασφαλείας. Ύστερα υπάρχει μια καθυστέρηση 5 δευτερολέπτων για να προλάβει ο χρήστης να πιάσει ταυτόχρονα τους SW4-SW1, διαβάζονται οι διακόπτες, αποθηκεύεται προσωρινά ο κωδικός ,μετά υπάρχει ακόμα μια καθυστέρηση 1 δευτερολέπτου και τέλος πρέπει να πατηθεί ο SW7 ώστε να αποθηκευτεί μόνιμα ο κωδικός και να συνεχίσουμε στο σύστημα ασφαλείας.

#### II)Σύστημα ασφαλείας:

Το σύστημα ασφαλείας ενεργοποιείται πρίζοντας τον SW0. Στην συνέχεια πρέπει να εισαχθεί ο κωδικός , ένα bit κάθε φορά ξεκινώντας από το SW4. Κάθε bit έχει 1,25 δευτερόλεπτα χρόνο για να πατηθεί ( $4 \times 1,25 = 5$  sec). Αν δεν πατηθεί κάποιο bit σε αυτό το χρονικό περιθώριο το σύστημα τον δέχεται ως μη πατημένο. Αν κανένας διακόπτης δεν πατηθεί το σύστημα καταλαβαίνει ότι πέρασαν 5 δευτερόλεπτα χωρίς εισαγωγή κωδικού (άρα ο κωδικός δεν μπορεί να είναι 1111). Τέλος αφού εισαχθεί ένας κωδικός γίνεται ο έλεγχος και πραγματοποιούνται οι ενέργειες που ζητάει η εκφώνηση αν είναι λάθος ή σωστός ο κωδικός και αν πατήθηκε 2 φορές λάθος. Αν πατήθηκε 2 φορές λάθος κωδικός τότε,αφού αναβοσβήσουν όλα τα LEDS με ρυθμό 1 sec για 5 sec συνολικά, εκτελείται μια RJMP που στέλνει την συνέχεια του προγράμματος σε ένα label στην τελευταία γραμμή του προγράμματος που οδηγεί στην εκτέλεση του προγράμματος πάλι από την αρχή.

#### II)Κανονική λειτουργία:

Στην κανονική λειτουργία περιέχονται πολλοί βρόχοι οι οποίοι διαβάζουν τους διακόπτες και εκτελούν όλες τις λειτουργίες που ζητούνται από την εκφώνηση.

### Γενικά σχόλια:

- Για την επεξεργασία του κωδικού ώστε να γίνουν σωστά οι εισαγωγές αυτού και η σύγκριση γίνεται με το συμπλήρωμα ως προς ένα. Συγκεκριμένα πρώτα γίνεται CLR στον καταχωρητή που θα αποθηκεύσει τον κωδικό που θα εισαχθεί στο σύστημα ασφαλείας, εισάγεται ένα bit κάθε φορά ,γίνεται το συμπλήρωμα ως προς ένα και προστίθεται σε αυτόν τον καταχωρητή. Μετά συγκρίνεται με τον καταχωρητή που περιέχει τον σωστό κωδικό ,στον οποίο έχει γίνει το συμπλήρωμα ως προς ένα.
- Όλα τα LEDS ανάβουν όπως ζητείται.
- Ο κώδικας περιέχεται σε 1 αρχεία .asm με σχόλια όπως αναφέρει μια ανακοίνωση στο ethmmy και όχι εδώ.

- Ο περισσότερος χρόνος αφιερώθηκε στις δυσκολίες που συναντήσαμε οι οποίες ήταν : να αποφασίσουμε ποιον κωδικό θα δέχεται το σύστημα , πως θα τον διαβάσει και να το εφαρμόσουμε στον κώδικα.Τελικά αποφασίσαμε να το κάνουμε σε 2 στάδια. Στο πρώτο εισάγεται με ταυτόχρονη πίεση των διακοπών ο κωδικός που θα δέχεται ως σωστό το σύστημα και στο δεύτερο ο κωδικός θα εισάγεται ένα bit την φορά.
- Η κύρια αποσφαλμάτωση έγινε στο AVR STUDIO 4 όπου χρησιμοποιήκαν breakpoints για τις ενοχές branch ώστε να λειτουργούν σωστά. Σε σχέση με την αναπτυξιακή κάρτα απλά ήταν πιο εύκολο και γρήγορο στην κάρτα να πιέζουμε τους διακόπτες από το να αλλάζουμε τιμές στον προσομοιωτή και χρειάστηκε να κάνουμε CLR στοον καταχωρητή R27 για να αποθηκεύσουμε τον κωδικό ένα bit κάθε φορά ,αφού η αποθήκευση γίνεται με πρόσθεση και οι καταχωρητές έχουν τυχαίες τιμές όταν ξεκινάει ο μικροελεγκτής.
- Βελτίωση στον κώδικα που τελικά αναπτύξαμε μπορεί γίνει αν συμπεριληφθεί και ο κωδικός 1111 ως αποδεκτός κωδικός ,αλλά τότε θα πρέπει να χρησιμοποιηθεί μια διαφορετική λογική στον κώδικα.