

# Μάθημα

## Εργαστηριακή Άσκηση 4

Λουδάρος Ιωάννης (1067400) - Χριστίνα Κρατημένου (1067495)



Μπορείτε να δείτε την τελευταία έκδοση του Project [εδώ](#) ή σκανάροντας τον κωδικό QR που βρίσκεται στην επικεφαλίδα.

## Περιγραφή Αναφοράς

Παρακάτω παραθέτουμε τις απαντήσεις μας στην “Εργαστηριακή Άσκηση 4” του μαθήματος “Προηγμένοι Μικροεπεξεργαστές” καθώς και σχόλια τα οποία προέκυψαν κατά την εκπόνηση της.

Σας ενημερώνουμε ότι λόγω θεμάτων συμβατότητας (Το Microchip Studio τρέχει μόνο σε intel επεξεργαστές), δεν ήμασταν σε θέση να ελέγξουμε κατά πόσο ο κώδικας μας είναι όντως λειτουργικός.

## Περιεχόμενα

Εισαγωγή.....	2
Πρώτο Ερώτημα .....	2
Δεύτερο και Τρίτο Ερώτημα .....	3

# Απαντήσεις

## Εισαγωγή

Για την καλύτερη αναγνωσιμότητα του κώδικα επιλέξαμε να δημιουργήσουμε ένα header file το οποίο περιέχει όλες τις συναρτήσεις, τις global σταθερές, καθώς και τα include statements που χρησιμοποιούμε. Μπορείτε να το βρείτε στον ίδιο κατάλογο με το παρόν έγγραφο, ως "iheader.h".

## Πρώτο Ερώτημα

```
1#include <iheader.h>
2
3
4//***** Main *****//
5
6int main(void)
7{
8    initialise_led();
9    initialise_switches();
10   set_TCA();
11   sei();
12
13   while (1)
14   {
15       if(access_granted==1)
16       {
17           start_TCA();
18       }
19   }
20 }
21
22
```

Η λογική σε αυτό το ερώτημα είναι απλή. Χρησιμοποιούμε τους διακόπτες 5 και 6 ως πληκτρολόγιο ώστε να εισάγουμε τον κατάλληλο κωδικό. Κάθε φορά που πατιέται ένας διακόπτης, ελέγχεται ποιος είναι αυτός (ο 5 ή ο 6) και συγκρίνεται με τον διακόπτη που θα έπρεπε να έχει πατηθεί ώστε να συμπληρωθεί το password. Η λειτουργικότητα αυτή υλοποιείται μέσω ενός interrupt, του `ISR(PORTF_PORT_vect)` και μιας συνάρτησης, της `check_password()`. Αφού συμπληρωθεί το password, χρησιμοποιούμε χρησιμοποιούμε έναν timer ώστε να δώσουμε χρόνο στον χρήστη να απομακρυνθεί και ύστερα ενεργοποιούμε τον συναγερμό.

```
1ISR(PORTF_PORT_vect)// Button Pressed
2{
3    // clear the interrupt flag
4    int intflags = PORTF.INTFLAGS;
5    PORTF.INTFLAGS = intflags;
6
7    // if SW5 is enabled
8    if((PORTF.INTFLAGS & 00100000)==1) current_digit=5;
9    // SW6 is enabled
10   else current_digit=6;
11
12   if (current_digit==password[checking_digit])
13   {
14       checking_digit++;
15   }
16   else
17   {
18       checking_digit=0;
19       wrong_tries++;
20   }
21   check_password();
22 }
```

Όπως φαίνεται ξεκάθαρα αριστερά. Το σύστημα περιμένει μέχρι η `check_password()`, να διαπιστώσει ότι ο κωδικός εισάχθηκε σωστά. Μόλις γίνει αυτό, δίνεται άδεια στον χρήστη να ξεκινήσει τον timer (`access_granted=1`).

```
1ISR(TCA0_CMP0_vect){
2    //clear the interrupt flag
3    int intflags = TCA0.SINGLE.INTFLAGS;
4    TCA0.SINGLE.INTFLAGS = intflags;
5
6
7    if (locked)
8    {
9        siren=1;
10       set_pwm();
11    }
12    else
13    {
14        locked=1;
15        disable_TCA();
16    }
17 }
18
19
```

Μόλις ο timer τελειώσει, μπαίνουμε στο `ISR(TCA0_CMP0_vect)`, όπου απενεργοποιείται ο TCA και σηματοδοτείται το κλείδωμα του συναγερμού (`locked=1`).

```
1void check_password()
2{
3    if (checking_digit==4)// This means that the first 4
4    digits (so all of them) were correct
5    {
6        access_granted=1;
7        checking_digit=0;
8        wrong_tries=0;
9        waiting_for_password=0;
10       siren=0;
11       PORTD.OUT |= PIN0_bm; //led is off
12
13       if (locked)
14       {
15           disable_TCA();
16           locked=0;
17           PORTD.OUT |= PIN0_bm; //led is off
18       }
19   }
20   else
21   {
22       access_granted=0;
23   }
24
25   if (wrong_tries>=3 && locked==1) set_pwm();
26 }
```

## Δεύτερο και Τρίτο Ερώτημα

Η λογική που προστίθεται σε αυτό το ερώτημα έχει να κάνει με τον εντοπισμό κίνησης από τον ADC.

Βλέπουμε ότι σε περίπτωση που ανιχνευθεί κίνηση (intruder=1) ενώ ο συναγερμός είναι κλειδωμένος (locked=1), τότε το σύστημα ξεκινάει το χρονόμετρο και περιμένει να δοθεί το password. Σε περίπτωση που ο χρήστης αποτύχει να δώσει το σωστό password μέσα στον επιτρεπτό χρόνο, τότε θέτουμε τον TCA0 σε PWM και ξεκινάμε την σειράνα, μέχρι να δοθεί σωστό password.

```
1#include <iheader.h>
2
3
4
5//***** Main *****//
6
7int main(void)
8{
9    initialise_led();
10   initialise_switches();
11   initialise_ADC();
12   free_running();
13   start_ADC();
14   set_TCA();
15   sei();
16
17   while (1)
18   {
19       if (access_granted==1 && locked==0) // 0 χρήστης
20       βάζει τον κωδικό του για να κλειδώσει το σπίτι του
21       {
22           start_TCA();
23       }
24
25       if (intruder==1 && locked==1)
26       {
27           if (!waiting_for_password)
28           {
29               PORTD.OUTCLR |= PIN0_bm; //LED is on
30               set_TCA();
31               start_TCA();
32               waiting_for_password=1;
33           }
34           else
35           {
36               while (!access_granted)
37               {
38               }
39               access_granted=0;
40               set_TCA();
41           }
42       }
43   }
44 }
45
46
```

```
1 // ADC Interrupt
2 ISR(ADC0_WCOMP_vect) //intruder found
3 {
4     int intflags = ADC0.INTFLAGS;
5     ADC0.INTFLAGS = intflags;
6     intruder = 1;
7
8 }
```

```
1void check_password()
2{
3    if (checking_digit==4)// This means that the first 4
4    digits (so all of them) were correct
5    {
6        access_granted=1;
7        checking_digit=0;
8        wrong_tries=0;
9        waiting_for_password=0;
10       siren=0;
11       PORTD.OUT |= PIN0_bm; //led is off
12
13       if (locked)
14       {
15           disable_TCA();
16           locked=0;
17           PORTD.OUT |= PIN0_bm; //led is off
18       }
19   }
20   else
21   {
22       access_granted=0;
23   }
24
25   if (wrong_tries>=3 && locked==1) set_pwm();
26 }
27
```