Συστήματα Μικρουπολογιστών

Εργαστηριακή Άσκηση 2

Ομάδα 12

Κυπριανίδης Αλέξανδρος-Χαράλαμπος 8012

Μάστορας Ραφαήλ-Ευάγγελος 7918

Εργασία

Το πρόγραμμα που ζητήθηκε να υλοποιήσουμε αφορούσε την δημιουργία ενός συστήματος ασφαλείας με την χρήση Assembly για μικροεπεξεργαστή AVR με το πρόγραμμα AVR studio 4. Αρχικά ορίσαμε μεταβλητές για την διευκόλυνση μας, που χρησιμοποιούμε σαν καταμετρητές και σαν «σημαίες», οι οποίες χρησιμοποιούνται για την ανίχνευση του πατήματος του εκάστοτε κουμπιού.

Αφού το πρόγραμμα αρχικοποιήσει τους καταχωρητές που είναι υπεύθυνοι για τα switches και τα ports, μέσα στο label **initPassword** εισάγεται ο επιθυμητός κωδικός. Αυτό επιτυγχάνεται είτε πατώντας συνεχόμενα τα πλήκτρα 0-3 είτε πατώντας διαδοχικά ένα πληκτρό για να καλύψουμε την περίπτωση επανάληψης ενός ψηφίου. Επειτά η αποθήκευση του κωδικού γίνεται πιέζοντας το πλήκτρο 7. Οι προδιαγραφές της υλοποίησης αυτής επιτρέπουν την χρήση μήκους κωδικού μέχρι 35 ψηφίων.Αυτό το μήκος μπορεί να μεταβληθεί αλλάζοντας τις θέσεις των διευθύνσεων της μνήμης **Data**, όπου και αποθηκεύεται ο κωδικός. Επίσης έγινε χρήση της στοίβας για την προσωρινή αποθήκευση των ψηφίων. Μετά την αποθήκευση του κωδικού έχουμε μεριμνήσει για την αναστολή του προγράμματος μέχρι να πατηθεί το πλήκτρο 0 και να μεταβεί το πρόγραμμα στο σύστημα ασφαλείας.

Το πρόγραμμα συνέχιζει στην ετικέτα **password_check**, όπου και μπορεί να εισαχθεί ο κωδικός. Με την επιλογή του πρώτου ψηφίου ξεκινά και μετράει ο χρόνος καθυστέρησης των 5 δευτερολέπτων, που έχει δοθεί σαν προδιαγραφή του σύστηματος. Η καθυστέρηση υλοποίεται με την χρήση κατάλληλου αριθμού βρόγχων επαναλήψεων. Με την χρήση στοίβας εισάγονται τα επιλεγμένα ψηφία και έπειτα συγκρίνονται αφού εξαχθούν από την στοίβα με τον κωδικό του συστήματος. Αυτό γίνεται στο label **check**.

Εφόσον δοθεί εσφαλμένος κωδικός το πρόγραμμα μεταβαίνει στο label **alarm1** και αναβοσβήνει το led0 με ρυθμό 1 sec. Για να επιτευχθεί ο ρυθμός χρησιμοποίησαμε βρόχους επαναλήψης με καθυστέρηση 0,5s. Η επαλήθευση του κωδικού γίνεται όπως περιγράφηκε προηγουμένως.

Εφόσον εισαχθεί ξανά λανθασμένος κωδικός το πρόγραμμα μεταβαίνει στην ετικέτα **alarm2**.

Και στις 2 παραπάνω περιπτώσεις αν εισαχθεί σωστά ο κωδικός το πρόγραμμα μεταβαίνει στην κανονική λειτουργία που ορίζεται από το label **normal_operation**.

Στο **alarm2** το πρόγραμμα μπαίνει σε ατέρμον βρόγχο όπου και αναβοσβήνουν όλα τα led με ρυθμό 1 sec.

Στο **normal_operation** το πρόγραμμα ανάβει το led1 και αναμένει το πάτημα κάποιου κουμπιού. Εφόσον πατηθεί το 1 και έπειτα πατηθεί το 0 ενεργοποίειται το σύστημα ασφαλείας ξανά. Αντίθετα, σε περίπτωση που πατηθεί ένα κουμπί εκ των 2-7, πριν πατηθεί το 1, το πρόγραμμα μεταβαίνει στην ανάλογη ετικέτα, όπου ανάβει το αντίστοιχο με το πλήκτρο led. Σε κάθε περίπτωση ανάβει και το led2.

Δυσκολίες

Την σημαντικότερη δυσκολία αντιμετωπίσαμε κατά την χρήση στοίβας καθώς επηρέαζε τα rcall και ret που υπήρχαν στο πρόγραμμα . Για τον λόγο αυτό χρειάστηκε να χρησιμοποιήσουμε rjmp και να δημιουργήσουμε τα αντίστοιχα label, για το μεγαλύτερο μέρος του προγράμματος.
Επίσης δυσκολευτήκαμε στο κομμάτι ταυτόχρονης πίεσης πλήκτρων για την εισαγωγή κωδικού καθώς και στο περιορισμό του χρόνου εισαγωγής κωδικού στο σύστημα ασφαλείας. Επιπλέον το αναβόσβημα των επιθυμιτών led με ρυθμό 1 sec χρειάστηκε πολλές δοκιμές.

Breakpoints

Για τον έλεγχο ορθής λειτουργίας του προγράμματος χρησιμοποιήσαμε breakpoints. Τα κυριότερα σημεία όπου χρησιμοποιήθηκαν breakpoints είναι κατά την εισαγωγή και τον έλεγχο του κωδικού, καθώς και στα σημεία όπου γίνεται ανάκτηση των ψηφίων από την στοίβα. Σε κάθε περίπτωση έπρεπε να ελέγχουμε τους καταχωρητές καθώς και τις επιλεγμένες θέσεις Data μνήμης.
Επιπρόσθετα τρέξαμε πολλές φορές των κώδικα μία-μία εντολή για το debugging.

Κώδικας του προγράμματος

```
.include "m16def.inc"
.org 0
.cseg
.def counter2=r30            ;metrhths pshfiwn ths ekastote prospa8eias eisagwghs kwdikou
.def counter=r28             ;metrhths  pshfiwn tou arxikou kwdikou
.def flag0=r22               ;deikths an path8hke to switch 0
.def flag1=r23
.def flag2=r24
.def flag3=r25
.def flag_alarm=r31          ;deikths oti path8hke estw kai ena switch ek twn 0-3 , wste na ksekinhsei to delay
.def delay_counter=r26       ;metrhths kyklwn brogxou epanalhpshs
sp_init:
        ldi r16,low(RAMEND)                 ;arxikopoihsh stack pointer
        out spl,r16
        ldi r16,high(RAMEND)
        out sph,r16
        ldi counter,1
switch_init:
        clr r17
        out DDRD,r17
        ser r17                             ;arxikopoihsh switch
        out PIND,r17
led_init:
        ser r17                             ;arxikopoihsh led
        out DDRB,r17
        ser r17
        out PORTB,r17
        rjmp initPassword
```

```
initPassword:                       ;eisagwgh arxikou kwdikou
        sbis PIND,0x00              ;if(Port D,pin0=0) then dont execute next command
        rjmp sw0_pressed            ;calls the sw0_pressed function
        ret0:
        sbis PIND,0x01              ;if(Port D,pin0=0) then dont execute next command
        rjmp sw1_pressed            ;calls the sw0_pressed function
        ret1:
        sbis PIND,0x02
        rjmp sw2_pressed
        ret2:
        sbis PIND,0x03
        rjmp sw3_pressed
        ret3:
        sbrc flag0,0                ;an to flag0 einai 0 pareleipse thn epomenh entolh
        rjmp sw0_unpressed
        ret4:
        sbrc flag1,0
        rjmp sw1_unpressed
        ret5:
        sbrc flag2,0
        rjmp sw2_unpressed
        ret6:
        sbrc flag3,0
        rjmp sw3_unpressed
        ret7:
        sbis PIND,0x07              ;an den exei path8ei to 7 pareleipse thn epomenh entolh
        rjmp sw7_pressed
        rjmp initPassword           ;brogxos epanalhpshs, pali apo thn arxh tou initPassword
sw0_pressed:
        sbrs flag0,0                ;elegxoume an an exei ksana mpei to programma sthn etiketa ayth
        ldi r18,0                   ;ka8ws trexei se kyklous kai emeis to kratame pathmeno
        sbrs flag0,0                ;an mpenei prwth for a tote bazoume ston kataxwrhth r18 to 0
        push r18                    ;bazoume sthn stoiba ton r18
        sbrs flag0,0
        adiw counter,1              ;ayksanoume ton metrhth
        ldi flag0,1                 ;kai 8etoume to flag0, 1 wste na ginetai o elegxos pou periegrapsa
        rjmp ret0                   ;epistrofh pisw
sw1_pressed:
        sbrs flag1,0
        ldi r18,1
        sbrs flag1,0
        push r18
        sbrs flag1,0
        adiw counter,1
        ldi flag1,1
        rjmp ret1
sw2_pressed:
        sbrs flag2,0
        ldi r18,2
        sbrs flag2,0
        push r18
        sbrs flag2,0
        adiw counter,1
        ldi flag2,1
        rjmp ret2
```

```
sw3_pressed:
        sbrs flag3,0
        ldi r18,3
        sbrs flag3,0
        push r18
        sbrs flag3,0
        adiw counter,1
        ldi flag3,1
        rjmp ret3
sw0_unpressed:
        sbic PIND,0x00          ;an ksepath8ei ena plhktro tote kanoume reset tis times, wste
        rjmp reset              ;na mporei na ksana path8ei. Etsi mporoume na baloume kwdiko
        rjmp ret4               ; me dyo h parapanw idia pshfia.
sw1_unpressed:
        sbic PIND,0x01
        rjmp reset
        rjmp ret5
sw2_unpressed:
        sbic PIND,0x02
        rjmp reset
        rjmp ret6
sw3_unpressed:
        sbic PIND,0x03
        rjmp reset
        rjmp ret7
reset:
        ldi flag0,0                     ;epanaprosdiorismos twn arxikwn timwn
        ldi flag1,0
        ldi flag2,0
        ldi flag3,0
        ldi flag_alarm,0
         ldi delay_counter,1
        ser r16                                 ;initialization of switches
        out PIND,r16
        rjmp initPassword
reset2:
        ldi counter2,1
        ldi flag0,0
        ldi flag1,0
        ldi flag2,0                     ;epanaprosdiorismos twn arxikwn timwn
        ldi flag3,0
        ldi flag_alarm,0
         ldi delay_counter,1
        ret




sw7_pressed:
        cpi counter,1               ;an exei path8ei to sw7 xwris na exei eisagx8ei kapoios kwdikos  prwta
        breq reset                  ;kaloume thn reset kai apo ekei pali thn initPassword
        sbis PIND,0x07              ; perimenoume mexri na ksepath8ei to sw7
        rjmp sw7_pressed
        clr r27                     ; xrhsh twn r27 kai r26 gia na kataxwrhsoume sthn DATA mnhmh
        ldi r26,$A0                 ;ton kwdikou pou eisax8hke
        mov r20,counter             ;metaferoume ton counter sto r20 giati xrishmopoioume allou ton  r18 kai
                                    ;epishs gia na meiwnoume ton ari8mo twn pshfiwn se ka8e epanalhpsh
```

```
                password_save:   ;brogxos gia osa pshfia exei to epi8ymito password
                pop r18              ;bgazoyme apo thn stoiba to teleytaio pshfio
                st -X,r18          ;apo8ykeysh toy sthn mnhmh data (8esh $A0-j,opou j ari8mos epanalhpshs)
                dec r20         ;meiwsh tou katametrhth kata 1
                cpi r20,1        ;an exoun apo8ykeytei ola ta pshfia proxwra alliws kalese thn password_save
                brne password_save
        clr r16
        out DDRD,r16
        ser r16                                 ;initialization of switches
        out PIND,r16
loop_sw0:
        sbic PIND,0x00                  ;perimenoune na path8ei to sw0
        rjmp loop_sw0
        ldi r16,1
        com r16
        out PORTB,r16
        clr r16
        out DDRD,r16
        ser r16                                 ;initialization of switches
        out PIND,r16
        rcall reset2                    ;epanaprosdiorismos twn arxikwn timwn
password_check:
        sbis PIND,0x00          ;an den exei path8ei to sw0 , pareleipse thn epomenh entolh
        rjmp check_sw0
        retc0:
        sbis PIND,0x01
        rjmp check_sw1
        retc1:
        sbis PIND,0x02
        rjmp check_sw2
        retc2:
        sbis PIND,0x03
        rjmp check_sw3
        retc3:
        clr r4
        cpse flag_alarm,r4          ;an to flagalarm einai 0  pareleipse thn epomenh entolh
        rjmp  delay025s1            ;ayto to kanoume gia na perimenei to programma mexri na path8ei to
        retdelay025s1:              ;prwto pshfio, kai meta na arxisei na metra 5 deyterolepta
        cpi delay_counter,21        ;an exoun perasei ta 5 deyterolepta kalese thn identify
        breq identify
        rjmp password_check         ;epanalhpsh
check_sw0:
        sbis PIND,0x00              ;perimenei mexri na ksepath8ei to sw0
        rjmp check_sw0
        ldi flag_alarm,1            ;8etei flag_alarm=1 , dhladh exei path8ei kapoio plhktro
        ldi r18,0                   ;me xrhsh stoibas kratountai proswrina ta pshfia pou eisagontai
        push r18
        adiw counter2,1               ;ayksanetai o katametrhths psifiwn2  kata 1
        rjmp retc0
check_sw1:
        sbis PIND,0x01
        rjmp check_sw1
        ldi flag_alarm,1
        ldi r18,1
        push r18
        adiw counter2,1
        rjmp retc1
```

```
check_sw2:
        sbis PIND,0x02
        rjmp check_sw2
        ldi flag_alarm,1
        ldi r18,2
        push r18
        adiw counter2,1
        rjmp retc2
check_sw3:
        sbis PIND,0x03
        rjmp check_sw3
        ldi flag_alarm,1
        ldi r18,3
        push r18
        adiw counter2,1
        rjmp retc3
identify:
        rjmp check              ;kalei thn synarthsh elegxoun kwdikwn
        goalarm1:
        mov counter,r21
        rcall reset2
        rcall alarm1
check:
        mov r21,counter         ;metaferoume ton counter wste na xrhsimopoihsoume afoba ton r18
        cp r21,counter2          ;an counter!=counter2 tote o kwdikos einai sigoura la8os
        brne goalarm1
        clr r27                  ;xrhsh toy zeygariou kataxwrhtwn r26-27 ( X) kai 28-29 (Y)
        ldi r26,$D0
        clr r29
        ldi r28,$A0
        mov r20,counter2     ;metaferoume ton counter2 wste na meiwnoume to r20 se ka8e epanalhpsh
        password_check1:
                pop r18
                st -X,r18
                ld r19,-Y
                cp r18,r19          ;sygkrish twn pshfiwn
                brne goalarm1       ;an de einai isa phgaine sto label goalarm1
                dec r20
                cpi r20,1            ;an den exoun elex8ei ola ta pshfia ksana kalese thn password_check1
                brne password_check1
        mov counter,r21          ;epanafora tou counter
        clr r21
        ser r16
        ldi r16,2
        com r16
        out PORTB,r16
        rcall normal_operation
normal_operation:                ;kanonikh leitourgia
        sbis PIND,0x02           ;an path8ei to sw2 kalese thn lightled 2
        rcall lightled2
        sbis PIND,0x03
        rcall lightled3
        sbis PIND,0x04
        rcall lightled4
        sbis PIND,0x05
        rcall lightled5
        sbis PIND,0x06
        rcall lightled6
```

```
        sbis PIND,0x07
        rcall lightled7
        sbis PIND,0x01          ;an path8ei to sw1 mpes sto atermon loop sw0, mexri na path8eito sw0
        rcall loop_sw0
        rjmp normal_operation

lightled2:
        ser r16
        ldi r16,4           ;4 giati 8eloume na anapsoume to led2 opote 00000100 =4
        com r16
        out PORTB,r16
        rjmp lightled2
lightled3:
        ser r16
        ldi r16,12           ; 00001100 giati 8eloume na anapsei to 2 kai to 3
        com r16
        out PORTB,r16
        rjmp lightled3
lightled4:
        ser r16
        ldi r16,20          ;10100 giati 8eloume na anapsei to 2 kai to 4
        com r16
        out PORTB,r16
        rjmp lightled4
lightled5:
        ser r16
        ldi r16,36          ;100100 giati 8eloume na anapsei to 2 kai to 5
        com r16
        out PORTB,r16
        rjmp lightled5
lightled6:
        ser r16
        ldi r16,68            ;1000100  giati 8eloume na anapsei to 2 kai to 6
        com r16
        out PORTB,r16
        rjmp lightled6
lightled7:
        ser r16
        ldi r16,132           ;100001000 giati 8eloume na anapsei to 2 kai to 7
        com r16
        out PORTB,r16
        rjmp lightled7


alarm1:                             ; periptwsh mias la8os prospa8ias
        mov r5,delay_counter
        lsr r5    ;to kanoume wste na doume an o r5 einai zhgos h artios
        lsl r5    ;ayto giati diairoume arxika me 2 kai meta pollaplasiazoume me 2 ton ari8mo
        rjmp offled             ;sbhnoume ta led
afteroffled:
        cpse delay_counter,r5
        rjmp onled                  ;ama einai monos anabei ta led
afteronled:                         ; ama einai zhgos paramenoun sbhsta ta led
        sbis PIND,0x00
        rjmp check1_sw0
        alarm1retc0:
        sbis PIND,0x01
        rjmp check1_sw1
```

```
        alarm1retc1:            ;oi gnwstoi mas elegxoi opws parapanw
        sbis PIND,0x02
        rjmp check1_sw2
        alarm1retc2:
        sbis PIND,0x03
        rjmp check1_sw3
        alarm1retc3:
        clr r4
        cpse flag_alarm,r4
        rjmp  delay05s1                 ;delay 0,5s gia 10 epanalhpseis
        retdelay05s1:
        cpi delay_counter,11
        breq identify2
        rjmp alarm1
check1_sw0:
        sbis PIND,0x00
        rjmp check1_sw0
        ldi flag_alarm,1
        ldi r18,0
        push r18                        ;omoia me tis prohgoumenes eisagwges pshfiwn pou eidame
        adiw counter2,1
        rjmp alarm1retc0
check1_sw1:
        sbis PIND,0x01
        rjmp check1_sw1
        ldi flag_alarm,1
        ldi r18,1
        push r18
        adiw counter2,1
        rjmp alarm1retc1
check1_sw2:
        sbis PIND,0x02
        rjmp check1_sw2
        ldi flag_alarm,1
        ldi r18,2
        push r18
        adiw counter2,1
        rjmp alarm1retc2
check1_sw3:
        sbis PIND,0x03
        rjmp check1_sw3
        ldi flag_alarm,1
        ldi r18,3
        push r18
        adiw counter2,1
        rjmp alarm1retc3
offled:
        ser r16
        out PORTB,r16     ;synarthsh poy sbhnei ta led
        rjmp afteroffled
onled:
        ser r16
        ldi r16,1
        com r16         ; synarthsh pou anabei to led1
        out PORTB,r16
        rjmp afteronled
```

```
identify2:
        rjmp check2     ;elegxos ths deyterhs prospa8ias
        goalarm2:
        mov counter,r21     ;metaferoume ton counter gia na xrhsimpoihsoume ton r18 eley8era
        rcall alarm2        ;an exoun apotyxei kai oi dyo prospa8ies kalese thn alarm2


check2:
        mov r21,counter
        cp counter,counter2
        brne goalarm2
        clr r27
        ldi r26,$D0
        clr r29                     ;omoia me to label check
        ldi r28,$A0
        mov r20,counter2
                password_check2:
                pop r18
                st -X,r18
                ld r19,-Y
                cp r18,r19
                brne goalarm2
                dec r20
                cpi r20,1
        brne password_check2
        mov counter,r21
        clr r21
        ser r16
        ldi r16,2
        com r16
        out PORTB,r16
        rcall normal_operation
alarm2:
        ser r16
        out PORTB,r16
        rjmp delay05s3;delay05s
        retdelay05s3:
        clr r16                     ;anaboun ta led gia 0.5s kai menoun sbhsta gia alla 0.5s
        out PORTB,r16
        rjmp delay05s4;delay05s
        retdelay05s4:
        rjmp alarm2
                    ;synarthseis delay poy xreiazontai
delay025s1:
  adiw delay_counter,1
   ldi  r21, 6
   ldi  r27, 19
   ldi  r29, 174
L1: dec  r29
   brne L1
   dec  r27
   brne L1
   dec  r21
   brne L1
   clr r21
   clr r27
   clr r29
   rjmp retdelay025s1
```

```
delay05s1:
   adiw delay_counter,1
   ldi  r21, 11
   ldi  r27, 38
   ldi  r29, 94
L2: dec  r29
   brne L2
   dec  r27
   brne L2
   dec  r21
   brne L2
   clr r21
   clr r27
   clr r29
   rjmp retdelay05s1

delay05s3:
   adiw delay_counter,1
   ldi  r21, 11
   ldi  r27, 38
   ldi  r29, 94
L3: dec  r29
   brne L3
   dec  r27
   brne L3
   dec  r21
   brne L3
   clr r21
   clr r27
   clr r29
  rjmp retdelay05s3
delay05s4:
   adiw delay_counter,1
   ldi  r21, 11
   ldi  r27, 38
   ldi  r29, 94
L4: dec  r29
   brne L4
   dec  r27
   brne L4
   dec  r21
   brne L4
   clr r21
   clr r27
   clr r29
   rjmp retdelay05s4
```