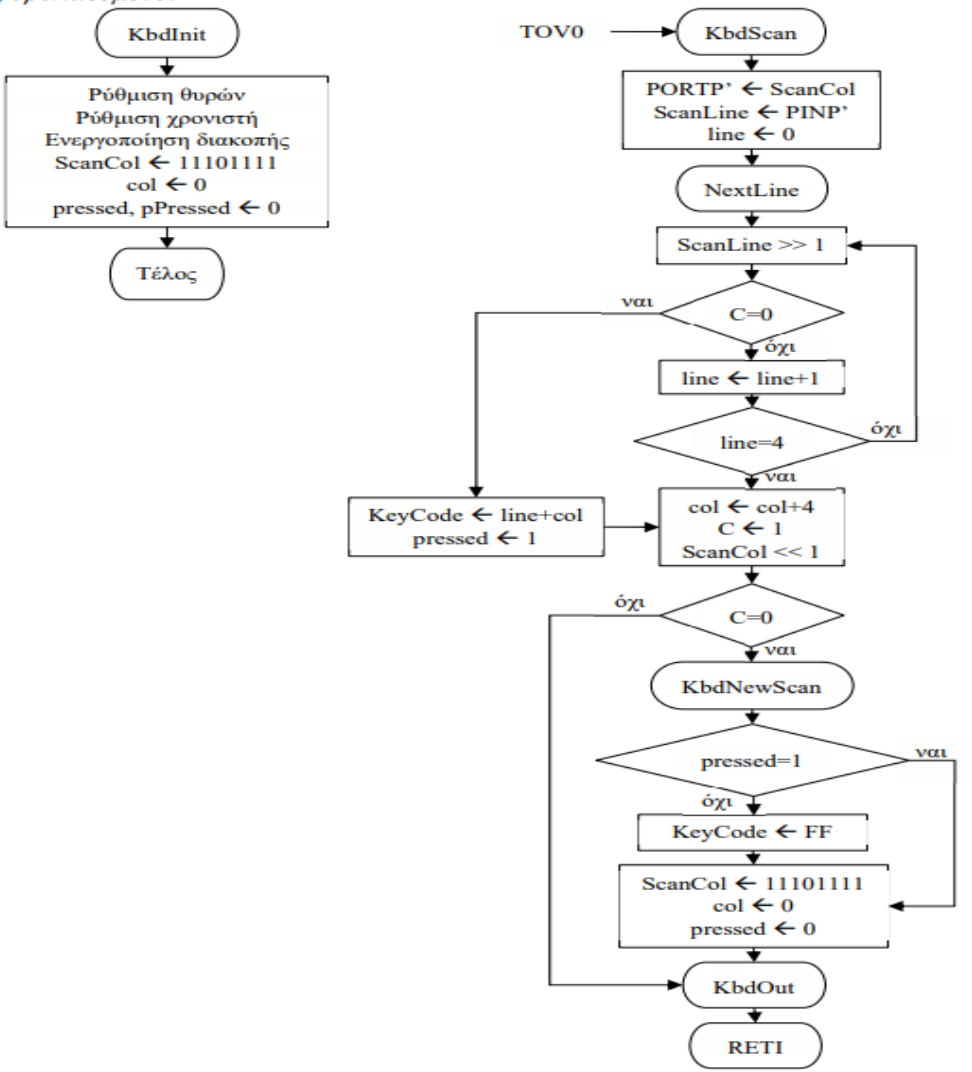
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΛΕΓΚΤΩΝ

ΑΣΚΗΣΗ 21

Τσιλιλής Κωνσταντίνος (143301)
Μωραϊτόπουλος Αλέξανδρος (052123)

ΑΣΚΗΣΗ 21.1

Κάντε κωδικοποίηση του διαγράμματος ροής υπολογισμού του κωδικού του πλήκτρου του σχήματος δημιουργώντας μια ρουτίνα διακοπής. Ενσωματώστε την στη ρουτίνα διακοπής της οθόνης με τους τρεις ενδείκτες επτά τμημάτων. Γράψτε ένα κυρίως πρόγραμμα που να εμφανίζει σε δύο ενδείκτες τον κωδικό του πλήκτρου που είναι κάθε φορά πιεσμένο.



Για την υλοποίηση της άσκησης αυτής θα χρησιμοποιήσουμε τον κώδικα της προηγούμενης άσκησης (20.3) και θα το παραμετροποιήσουμε στα δεδομένα της καινούριας άσκησης. Οι νέες μεταβλητές που θα χρησιμοποιήσουμε είναι οι:

```
.include "m32def.inc"

.def scancol=R17
.def work=R16
.def workl=R16
.def workh=R18
.def font=R19
.def counter=R20
.def line =R21
.def column= R22
.def pressed=R23
.def pPressed=R12
.def scanline=R13
.def KeyCode=R24
.def KeyValue=R14

.equ segport=PORTA
.equ scanport=PORTB

.dseg
fontBuf: .byte 5
fontPointer: .byte 2
someInt: .byte2
.cseg
```

Ο νέος μας κώδικας φαίνεται στην επόμενη σελίδα.

```
.include "m32def.inc"

.def scancol=R17
.def work=R16
.def workl=R16
.def workh=R18
.def font=R19
.def counter=R20
.def line =R21
.def colume= R22
.def pressed=R23
.def pPressed=R12
.def ScanLine=R13
.def KeyCode=R24
.def KeyValue=R14

.equ segport=PORTA
.equ scanport=PORTB

.dseg
fontBuf: .byte 5
fontPointer: .byte 2

.cseg

.org 0
rjmp reset

.org OVF0addr
rjmp dispscan

reset:
ldi work,high(RAMEND)
out SPH,work
ldi work,low(RAMEND)
out SPL,work

dispinit:
ldi work,0b11110000
out scanport-1,work ; DDRD pins7-4 exodoi pins 3-0 eisodoi
ser work ;set register kanei 1 ton kataxoriti
out scanport,work ;portb eisodoi me pull up
out segport-1,work ;ddra exodoi

ldi scancol,0b11101111 ;   arxise tin sarwsh apo ton proto

ldi work,low(fontBuf)
sts fontPointer,work
ldi work,high (fontBuf)
sts fontPointer+1,work

ldi work,(1<<TOIE0)
out TIMSK,work
ldi work,(1<<CS02) ;prescaler 256
out TCCR0,work

KBDinit:
clr colume
clr pressed
clr pPressed


SEI

main:
mov work,keyCode
call prn1hex
rjmp main


dispscan:
push work
in work,SREG
push work
;sozoume ton
push XH
push XL

ldi work,0b11111111
out scanport,work ;svinoume ta anamena psifia

lds XL,fontPointer
lds XH,fontPointer+1 ;diavazoume ton fontpointer apo tin sram
ld work,X ;emesh fortosh
sbiw X,1 ;meiwnoume ton x kata ena gia to epomeno pass
sts fontPointer+1,XH
sts fontPointer,XL ;store direct to ram

out segport,work
out scanport,scancol; anavoume to tmima pou exei seira



dispscan0:
nop


KBDscan:
in scanline,scanport-2 ;pin4 thira a
clr line

NextLine:
ror Scanline
brcc keyFound
inc line
cpi line,4
brne NextLine
rjmp keyNotFound
```

```
keyNotFound:
subi colume,-4
sec
rol ScanCol
brcs KBDout

KBDNewScan:
cpi pressed,1
breq KBDNewScan2
ser KeyCode

KBDNewScan2:
ldi ScanCol,0b11101111
clr colume
clr pressed
ldi work,high(fontBuf+3)
sts fontPointer+1,work
ldi work,low(fontBuf+3)
sts fontPointer,work

KBDOut:
pop XL
pop XH
pop work
out SREG,work
pop work
RETI


PRN1hex:
push work
andi work,0x0F
mov R2,work
call getfont
sts fontBuf,font

pop work
push work
swap work
andi work,0x0F
mov R2,work
call getfont
sts fontBuf+1,font

ser work
sts fontbuf+2,work

pop work
ret
```

```
getfont :
sub font,font
ldi ZL,low(FontTable<<1)
ldi ZH,high(FontTable<<1)
add ZL,R2
adc zh,font
lpm font,z
ret


FontTable:
.db 0b00100010,0b10101111,0b00110001,0b00100101 ;0-3
.db 0b10101100,0b01100100,0b01100000,0b00100000;4-7
.db 0b00100000,0b00100100,0b00101000,0b11100000;8-b
.db 0b01110010,0b10100001,0b01110000,0b01111000;c-f
```

ΑΣΚΗΣΗ 21.2

Σε αυτή την άσκηση μας ζητείται να δημιουργήσουμε μία υπορουτίνα “GetKeyValue”, η οποία θα υπολογίζει την αξία του πλήκτρου που πατάμε. Η λογική της είναι παρόμοια με την υπορουτίνα “getfont” που έχουμε ήδη δημιουργήσει, δηλαδή θα φτιάξουμε πάλι ένα πίνακα που θα αποθηκεύονται οι αξίες των πλήκτρων, οι οποίες θα αυξάνονται 1 ανά γραμμή και 4 ανά στήλη.

```
KeyTable:
.db 1,4,7,14
.db 2,5,8,0
.db 3,6,9,15
.db 10,11,12,13
```

Η υπορουτίνα “GetKeyValue” που δημιουργήσαμε θα πρέπει να τοποθετεί τον Z στην αρχή του πίνακα “KeyTable”. Κάθε φορά που πατάμε κάποιο πλήκτρο, προσθέτουμε στον Z ένα KeyCode, με το οποίο θα βρούμε την αξία του πλήκτρου μας (δηλ. Z = 0 + KeyCode). Ο κώδικας της ρουτίνας μας είναι ως εξής:

```
GetKeyValue:
sub KeyValue,KeyValue
ldi ZL,low(KeyTable<<1)
ldi ZH,high(KeyTable<<1)

add ZL,KeyCode
adc ZH,KeyValue
lpm KeyValue,Z
ret
```

LINKS ΒΙΝΤΕΟ ΑΣΚΗΣΕΩΝ

- Άσκηση 21_1 :https://youtu.be/5JuDM6ahAsM
- Άσκηση 21_2 :https://youtu.be/1Lkl7IJSQMw
- Άσκηση 21_3 :