

Εργαστήριο Μικροϋπολογιστών

6η Εργαστηριακή άσκηση

Ομάδα Γ04

Συνεργάτες:

- Σκούλος Ραφαήλ Α.Μ: 03112404
- Αναστάσης Σταθόπουλος Α.Μ: 03112140
- Τζίνης Ευθύμιος Α.Μ: 03112007

Άσκηση 1

```
.include "m16def.inc"
```

```
.org 0x0
```

```
rjmp reset
```

```
.org 0x4
```

```
rjmp ISR1
```

```
reti
```


ISR1:

check:

```
ldi r21 ,(1 << INTF1)
out GIFR ,r21                ;set 0 the bit7 of GIFR
ldi r24 , low(5)
ldi r25 , high(5)           ;load r25:r24 with 5
rcall wait_msec              ;delay = 5ms
in r21,GIFR
andi r21,128
cpi r21,0                    ;if bit7 of GIFR==0 check again
brne check
cli                          ;deactivate interrupts
push r26                     ;save r26 and SREG
in r26,SREG
push r26

in r28,PIND
andi r28,128
cpi r28,128
brne not_incr
inc r27                      ;increase the interrupt counter
```

not_incr:

```
out PORTA,r27
pop r26
out SREG,r26
pop r26                      ;pop back r26,SREG
sei                          ;activate interrupts
```

reti

;delay routines

wait_usec:

sbiw r24,1	;2 circle (0.250 μ sec)
nop	;1 circle (0.125 μ sec)
nop	
nop	
nop	
brne wait_usec	;1 or 2 circles (0.125 or 0.250 μ sec)
ret	;4 circles (0.500 μ sec)

wait_msec:

push r24	;2 circles (0.250 μ sec)
push r25	
ldi r24,low(998)	;1 circle
ldi r25,high(998)	
rcall wait_usec	;3 circles (0.375 μ sec)
pop r25	;2 circles
pop r24	
sbiw r24 , 1	;2 circles
brne wait_msec	;1 or 2 circles
ret	;4 circles

Άσκηση 2

```
.include "m16def.inc"
```

```
.def temp=r17
```

```
.def cnt=r18
```

```
.def loops=r19
```

```
.def init=r23
```

```
.def i=r22
```

```
.org 0x0
```

```
rjmp reset
```

```
.org 0x2
```

```
rjmp ISR0
```

```
reset:
```

```
    ldi r24,low(RAMEND)                ;initialize stack pointer
```

```
    out SPL,r24
```

```
    ldi r24,high(RAMEND)
```

```
    out SPH,r24
```

```
    clr r26
```

```
    out DDRA,r26
```

```
    ser r26
```

```
    out DDRB , r26                    ;initialize PORTB as output
```

```
    out DDRC,r26                      ;initialize PORTC as output
```

```
    clr r26                           ;initialize the counter
```

```
    ldi r24 ,( 1 << ISC01) | ( 1 << ISC00) ;make INT0 happen at
```

```
    out MCUCR , r24                  ;the rising edge of the signal
```

```
    ldi r24 ,( 1 << INT0)            ;set INT0 in interrupt mask
```

```

    out GICR , r24

    sei                                ;enable interrupt

loop1:

    out PORTB , r26                    ;show the counter at PORTA

    ldi r24 , low(100)                 ;load r25:r24 with 100

    ldi r25 , high(100)                ;delay = 100ms

    rcall wait_msec

    inc r26                            ;increase counter

    rjmp loop1                         ;repeat


ISR0:

    ldi r24 ,(1 << INTF0)

    out GICR ,r24                      ;set 0 the bit6 of GICR

    ldi r24 , low(5)

    ldi r25 , high(5)                  ;load r25:r24 with 5000

    rcall wait_msec                    ;delay = 5ms

    in r21,GICR

    andi r21,0x40

    cpi r21,0                          ;if bit6 of GICR==0 check again

    brne ISR0

    cli

    push r26

    in r26,PINA                        ;save the content of r26 and SREG

    mov init,r26

    in r26,SREG

    push r26

    ldi i,1

    ldi cnt,0                          ;initialize counter of 0

```

ldi loops,0 ;initialize number of loops,must be done 8 loops to check all the bits

counter:

mov temp,init

and temp,i

cp temp,i ;check if the i-th bit of the number of PORTA == 1

brne dontinc

inc cnt ;if so increase the counter

dontinc:

inc loops

;increase loops

lsl i ;shift i left

cpi loops,8

brlo counter ;if loops are less than 8 ,loop again

ldi r20,0x00

cpi cnt,0

breq ok ;if counter is 0 don't turn any led on

seton: lsl r20 ;if not turn on the appropriate leds

inc r20

dec cnt

cpi cnt,0

brne seton

ok: out PORTC,r20

ldi r24 , low(998)

ldi r25 , high(998)

rcall wait_msec ;delay = 1s

pop r26

out SREG,r26

pop r26

sei

reti

wait_usec:

sbiw r24 ,1	;2 cycle (0.250 µsec)
nop	;1 cycle (0.125 µsec)
nop	
nop	
nop	
brne wait_usec	;1 or 2 cycles (0.125 or 0.250 µsec)
ret	;4 cycles (0.500 µsec)

wait_msec:

push r24	;2 cycles (0.250 µsec)
push r25	
ldi r24 , low(998)	;1 cycles
ldi r25 , high(998)	
rcall wait_usec	;3 cycles (0.375 µsec)
pop r25	;2 cycles
pop r24	
sbiw r24 , 1	;2 cycles
brne wait_msec	;1 or 2 cycles
ret	;4 cycles

Άσκηση 3

```
.include "m16def.inc"

;suwnotita tou EasyAVR6 = 8MHz ara
;auksisis tou TCNT1 = 8MHz/1024=7812.5Hz

.equ Hvalue=high(49911) ;thelw yperxeilish meta apo 2 sec ara
;12*7812.5=15625 kyklous

.equ Lvalue=low(49911) ;ara arxikh timh 65536-15625 = 49911 = 0xC2F7

.def flag=r17 ;xrsh flag gia na kserw an mpainw

; ksana stin rutina eksuphrethshs ths diakophs
;wste na anapsw ola ta LEDs tou PORTB

.org 0x00
rjmp reset
reti

.org 0x04
rjmp ISR1
reti ;rutina eksupuretisis INT1

.org 0x010
rjmp TIMER1 ;rutina eksupiretisis tis diakopis uperxeilisis tou timer1
reti

reset:

ldi r26,low(RAMEND) ;initialize stack pointer
out SPL,r26

ldi r26,high(RAMEND)
out SPH,r26

ser r26

out DDRB,r26 ;arxikopoioume to PORTB gia eksodo
```

```

clr r26

out DDRD,r26                ;arxikopoioume to PORTD gia eisodo
out DDRA,r26                ;arxikopoioume to PORTA gia eisodo

ldi r23,( 1 << ISC11) | ( 1 << ISC10)
out MCUCR,r23                ;diakoph INT1 se shma thetikhs akmhhs
ldi r23,( 1 << INT1)        ;orizoume tin diakoph INT1
out GICR,r23

ldi r23 ,(1<<TOIE1)          ;energopoihsh diakophs yperxeilishs tou TCNT1
out TIMSK ,r23

ldi r23 ,(1<<CS12) | (0<<CS11) | (1<<CS10) ;syxnotita auksisis xronisth CLK/1024
out TCCR1B ,r23

ldi flag,0
sei                          ;energopoihsh diakopwn

```

;Main Program

```

loop1:
    in r27,PINA
    andi r27,128
    cpi r27,128
    breq ready                ;elegxw an htan PA7=1
    rjmp loop1

ready:
    in r27,PINA
    andi r27,128
    cpi r27,128                ;kai me ta egine PA7=0

```

breq ready

;diladi an egine PUSH to PA7

call ISR1

;kai tote exw interrupt

rjmp loop1

;Routina eksuphrethshs INT1

ISR1:

check:

ldi r23 ,(1 << INTF1)

out GIFR ,r23 ;set 0 the bit7 of GIFR

ldi r24 , low(5)

ldi r25 , high(5) ;load r25:r24 with 5

rcall wait_msec ;delay = 5ms

in r21,GIFR

andi r21,128

cpi r21,0 ;if bit7 of GIFR==0 check again

brne check

ldi r23,Hvalue ;Arxikopoihsh tou TCNT1

out TCNT1H,r23 ;gia uperxeilisi meta apo 2sec

ldi r23,Lvalue

out TCNT1L,r23

cpi flag,0

breq first_time ;an flag=0 dn exei ksanaginei interrupt(sta 2 sec pou metraw)

ldi flag,1

ldi r26,255

out PORTB,r26

ldi r24,low(500) ;r25:r24 = 200

ldi r25,high(500)

rcall wait_msec ;delay 0.5 sec

first_time: ;prwth fora pou eGINE interrupt (sta 2 sec pou metraw kathe fora)

ldi flag,1 ;flag = 1 giati an ksanaexw interrupt tha xreiatei na

ldi r26,1 ; anapsw ola ta LEDs tou PORTB

out PORTB,r26 ;anamma tou PB0

reti

;rutina eksupiretisis yperxeilishs xronisth

TIMER1:

ldi flag,0 ;epanafora tou flag sto 0

ldi r26,0

out PORTB,r26 ;mhdenismos tou PB0

reti

;dealay routines

wait_usec:

sbiw r24,1 ;2 circle (0.250 μ sec)

nop ;1 cilrcle (0.125 μ sec)

nop

nop

nop

brne wait_usec ;1 or 2 circles (0.125 or 0.250 μ sec)

ret ;4 circles (0.500 μ sec)

wait_msec:

push r24	;2 circlces (0.250 μ sec)
push r25	
ldi r24,low(998)	;1 circlce
ldi r25,high(998)	
rcall wait_usec	;3 circles (0.375 μ sec)
pop r25	;2 circlces
pop r24	
sbiw r24,1	;2 circles
brne wait_msec	;1 or 2 circles
ret	;4 circles