

# Κεφάλαιο 20



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

Τσιλιλής Κωνσταντίνος  
143301  
Μωραϊτόπουλος Αλέξανδρος  
052123

## Άσκηση 19.1

Δοκιμάστε το πρόγραμμα μέτρησης της περιόδου μιας παλμοσειράς. Στη συνέχεια απαντήστε στις παρακάτω ερωτήσεις.

Α) Πως μπορούμε να το άνω άκρο της περιοχής μέτρησης χωριάς να μειώσουμε την ακρίβεια; (δηλαδή με ίδιο διαιρέτη)

Μπορούμε να αυξήσουμε το άνω άκρο της περιοχής μέτρησης χωρίς να μειώσουμε την ακρίβεια προσθέτοντας στη ρουτίνα διακοπής μας ένα Software Timer (stimer). Καλό θα είναι όμως να το κάνουμε αυτό εφόσον μπορούμε να διαχειριστούμε την επιβάρυνση που θα επιφέρει στο πρόγραμμα. Γνωρίζουμε ότι το άνω άκρο της περιοχής μέτρησης του T1 είναι 65535 παλμοί διαιρέτη, οπότε αν στην αρχή του προγράμματος δηλώσουμε τον stimer μας με τιμή 2 και τον συμπεριλάβουμε στη ρουτίνα διακοπής, καταφέρουμε ουσιαστικά να διπλασιάσουμε το άνω άκρο της περιοχής μέτρησης.

Έτσι τώρα το πρόγραμμα θα μπαίνει στη ρουτίνα διακοπής, θα μειώνει τον stimer και όταν μηδενίζει θα εκτελεί τις λειτουργίες που μας ενδιαφέρουν, θα ξαναπέρνει αρχική τιμή και θα συνεχίζει. Στο σχήμα 1.1 φαίνεται η ρουτίνα διακοπής.

```
IntCapRout:                ; Routina diakopis
    DEC stimer
    BRNE return
    IN IntWorkL,SREG
    PUSH IntWorkL
    IN CurCaptL,ICR1L      ; Otan diavazoume 16-bitο kataxwriti prwta diavazoume to low byte
    IN CurCaptH,ICR1H      ; Meta to high byte
    MOV PeriodL,CurCaptL   ; Bazoume sto periodlow to current capture low
    MOV PeriodH,CurCaptH   ; Bazoume sto peridhigh to current capture high
    SUB PeriodL,PrevCaptL  ; Afairoume apo to current capture low to previous capture low
    SBC PeriodH,PrevCaptH  ; Afairoume apo to current capture high to previous capture high
    MOV PrevCaptL,CurCaptL ; Previous capture low <- current capture low
    MOV PrevCaptH,CurCaptH ; Previous capture high <- current capture high
    POP IntWorkL
    OUT SREG,IntWorkL
return:
    RETI
```

Σχήμα 1.1

Γ) Ποιός είναι ο περιορισμός στο κάτω άκρο της περιοχής μέτρησης (ελάχιστη περίοδος) που εισάγεται από την ρουτίνα διακοπής; Βρείτε την περιοχή μέτρησης για μέγιστο σφάλμα 1%.

Η διάρκεια εκτέλεσης της ρουτίνας διακοπής είναι συνολικά 20 παλμοί  $c_{pu}$ . Στο παρακάτω πίνακα (σχ 1.2) φαίνεται η διάρκεια εκτέλεσης των εντολών της ρουτίνας διακοπής, όπου με  $n$  συμβολίζεται το πλήθος των εντολών :

Εντολή	$n$	Καταστάσεις	Θέση
RJMP	1	2	Στο διάνυσμα της διακοπής
IN	3	1	Στη ρουτίνα διακοπής
PUSH	1	2	Στη ρουτίνα διακοπής
MOV	4	1	Στη ρουτίνα διακοπής
SUB	1	1	Στη ρουτίνα διακοπής
SBC	1	1	Στη ρουτίνα διακοπής
POP	1	2	Στη ρουτίνα διακοπής
OUT	1	1	Στη ρουτίνα διακοπής
RETI	1	4	Στη ρουτίνα διακοπής

Σχήμα 1.2

Έχουμε  $F_{cpu} = 16_{MHz}$  οπότε  $\frac{1}{16 \cdot 10^6} = 1,25 \mu s$  Άρα η ελάχιστη περίοδος που μπορεί να μετρηθεί βάση της “καθυστέρησης” της ρουτίνας διακοπής είναι 1.25 $\mu s$ .

## ΑΣΚΗΣΗ 19.2

Δοκιμάστε το πρόγραμμα μέτρησης της συχνότητας μιας παλμοσειράς. Υπολογίστε την περιοχή μέτρησης για μέγιστο σφάλμα 1% .

Το άνω άκρο της περιοχής μέτρησης είναι  $F_{cpu} \times T_p$ . Εδώ έχουμε prescaler 1024 και  $T_p = 64$ . Οπότε το άνω άκρο της περιοχής μέτρησης είναι  $F_{cpu} \times T_p = 16_{MHz} \times 64 = 1024_{ms}$ . Σε κάθε περίπτωση η συχνότητα

εισόδου δεν μπορεί να είναι μεγαλύτερη από  $F_{cru} \times T_p = 1024\text{MHz}$ . Το κάτω άκρο της περιοχής μέτρησης εξαρτάτε από το απόλυτο σφάλμα. Για ακρίβεια 1% και απόλυτο σφάλμα :

$$\frac{16_{MHz}}{1024} = 15.6ms$$

Για ακρίβεια 1%  $\frac{x}{15.6 \times 10^{-3}} = 1\% = 156\mu s$

Άρα για prescaler 1024 και ακρίβεια 1% η περιοχή μέτρησης είναι από

6,410Khz μέχρι 1024MHz .

Στο σχήμα 1.3 φαίνεται ο κώδικας της άσκησης

```
.include "a32def.inc"
.def work=R16
.def IntWork1=R17 :Use Only T1 interrupt
.def IntWork0=R18 :Use Only T0 interrupt
.def CounterL=R19 :the middle byte of cycle counter
.def CounterH=R20 :the high byte of cycle counter
.def FreqL=R4 : low byte 3-bit frequency counter
.def FreqH=R5 : medium byte 3-bit frequency counter
.def FreqH=R6 : high byte 3-bit frequency counter
.org 0
RJMP setup
.org OC1Aaddr
RJMP t1compA : Dianysa diakopis sugkrisis A
.org OV0addr
RJMP t0ovf
setup:
LDI work, low(RAMEND)
OUT SPL, work
LDI work, high(RAMEND)
OUT SPH, work
LDI work, 0b00000000 :PB.0=T0 clock input
OUT DDRB, work
LDI work, high(15624) :15625 palaoi P gia 1s
OUT OC1AH, work
LDI work, low(15624)
OUT OC1AL, work
CLR CounterL : Mhdenizoume ta 2 high bytes tou Freq
CLR CounterH
LDI work, (1<<TOIE0)|(1<<OCIE1A) : Energopoihsh diakopwn
OUT TIMSK, work
LDI work, (1<<CS02)|(1<<CS01)|(1<<CS00)|(0<<WGM01)|(0<<WGM00)
OUT TCCR0, work : Vazei apros T0 se Mode 0 se eksvteriko roloi ston akrodekti Tx (anerxomeni parufi)
LDI work, (0<<WGM11)|(0<<WGM00) : T1 se Mode 4
OUT TCCR1A, work
LDI work, (1<<CS12)|(1<<CS10)|(1<<WGM12)
OUT TCCR1B, work
SEI
loop:
NOP
NOP
NOP
RJMP loop
t0ovf:
IN IntWork0, SREG
PUSH IntWork0
INC CounterL : To medium byte tou metriti mas
BRNE t0ovfout
INC CounterH : To high byte tou metriti mas
t0ovfout:
POP IntWork0
OUT SREG, IntWork0
RETI
t1compA:
IN FreqL, TCNT0 : Diavase to metriti palawn T0, o opoios einai kai to low byte tou metriti mas
MOV FreqH, CounterL : Filaxe to low byte tou T1 sto FreqH
MOV FreqH, CounterH : Filaxe to high byte tou T1 sto FreqH
LDI IntWork1, 0 : De xalaxe tis shmaies
OUT TCNT0, IntWork1 : Mhdenizoume to metriti tou T0, ara kai to low byte tou metriti mas
LDI CounterL, 0
LDI CounterH, 0 : Mhdenizoume ta ypoloipa 2 byte tou metriti
RETI
```

Σχ. 1.3

Στην setup επιλέγουμε T0 σε κανονική λειτουργία (WGM bits =0) και Prescaler 64 (CS01 και CS00 =1). T1 σε CTC λειτουργία (WGM12=1) και prescaler 1024 (CS12 και CS10 =1). Έχουμε έτσι ξεκινήσει τους χρονιστές.