# Άσκηση 3Γ. Μετρητής δεκάτων δευτερολέπτου σε 3 ενδείκτες επτά τομέων

**Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C για τον μικροελεγκτή PIC 18F4550 με το οποίο ο μικροελεγκτής λειτουργεί ως χρονόμετρο με τρεις ενδείκτες 7 τομέων κοινής καθόδου όπου εμφανίζονται δέκατα του δευτερολέπτου.**

**Ο χρόνος αρχίζει να μετρά σε δέκατα του δευτερολέπτου όταν ο ακροδέκτης RD0 τεθεί σε λογικό 0 και για όσο χρόνο είναι σε λογικό 0.**

**Αν η ένδειξη χρόνου φθάσει τα 999 δέκατα του δευτερολέπτου συνεχίζεται η μέτρηση από το 000.**

**Αν αλλάξει η κατάσταση του ακροδέκτη RD0 από λογικό 0 σε λογικό 1 ο ενδείκτης σταματάει να κυλάει και δείχνει την τιμή του χρόνου κατά την στιγμή της αλλαγής. O χρόνος συνεχίζει να μετράει αλλά δεν εμφανίζεται στους ενδείκτες.**

**Αν ο χρήστης αλλάξει την κατάσταση του RD0 από λογικό 1 σε λογικό 0 τότε εμφανίζεται και πάλι η τρεχούμενη ένδειξη του χρόνου στους ενδείκτες.**

**Η παραπάνω διαδικασία αλλαγής κατάστασης με την βοήθεια του ακροδέκτη RD0 μπορεί να επαναλαμβάνεται. Να χρησιμοποιηθεί η μέθοδος της διακοπής από υπερχείλιση του timer0 για την μέτρηση του χρόνου και πίνακες μετατροπής για την οδήγηση των ενδεικτών.**

**Συνοπτικά:**

***RD0 1🡪0 :*** *Ξεκινάει ο μετρητής δεκάτων του δευτερολέπτου.*

***RD0 0:*** *Εμφανίζεται στους ενδείκτες η τρεχούμενη τιμή της μέτρησης χρόνου(σε δέκατα του δευτερολέπτου)*

***RD0 1 :*** *Εμφανίζεται η τιμή του μετρητή στους ενδείκτες κατά την στιγμή μετάβασης από 0 σε 1*

*(Ο μετρητής χρόνου συνεχίζει να μετράει αλλά δεν φαίνεται στους ενδείκτες)*

***RD0 0:*** *Εμφανίζεται ξανά στους ενδείκτες η τρεχούμενη τιμή της μέτρησης χρόνου(σε δέκατα του δευτερολέπτου)*

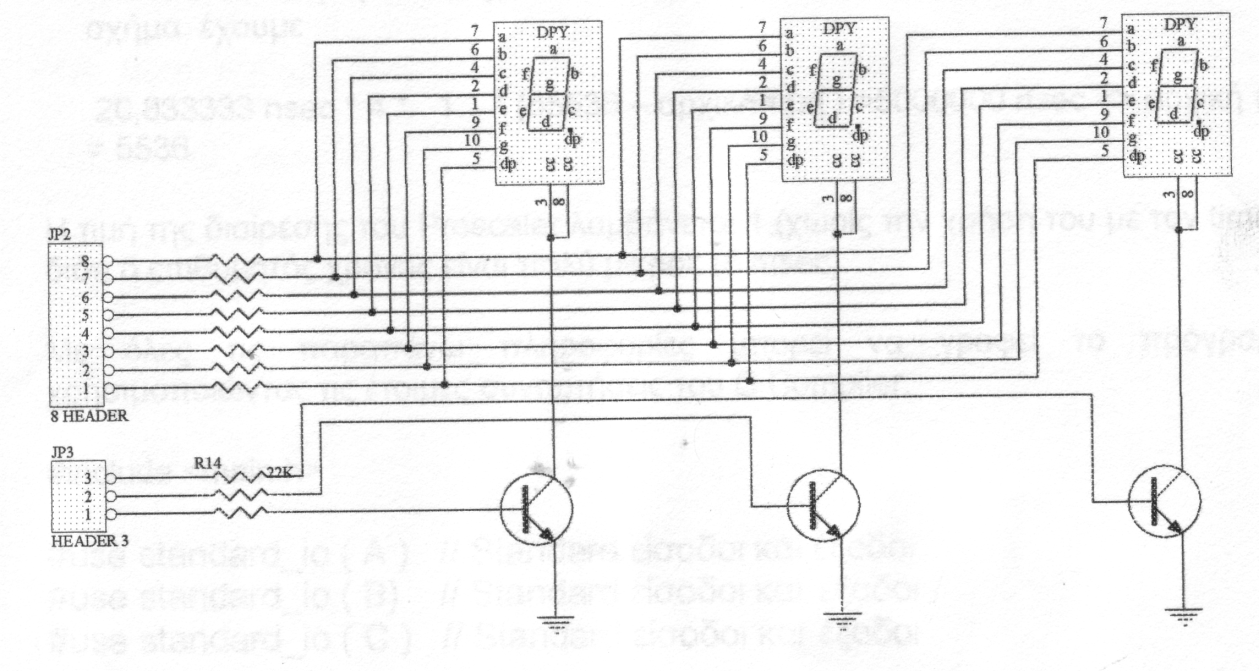
***RD0 1 :*** *Εμφανίζεται η τιμή του μετρητή στους ενδείκτες κατά την στιγμή μετάβασης από 0 σε 1*

*(Ο μετρητής χρόνου συνεχίζει να μετράει αλλά δεν φαίνεται στους ενδείκτες)*

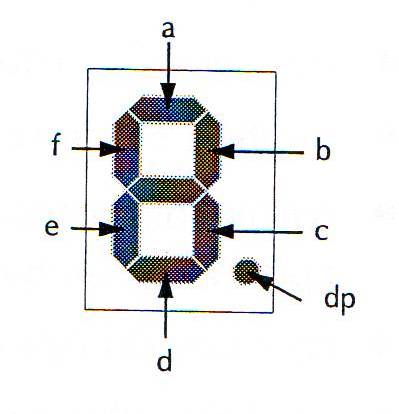
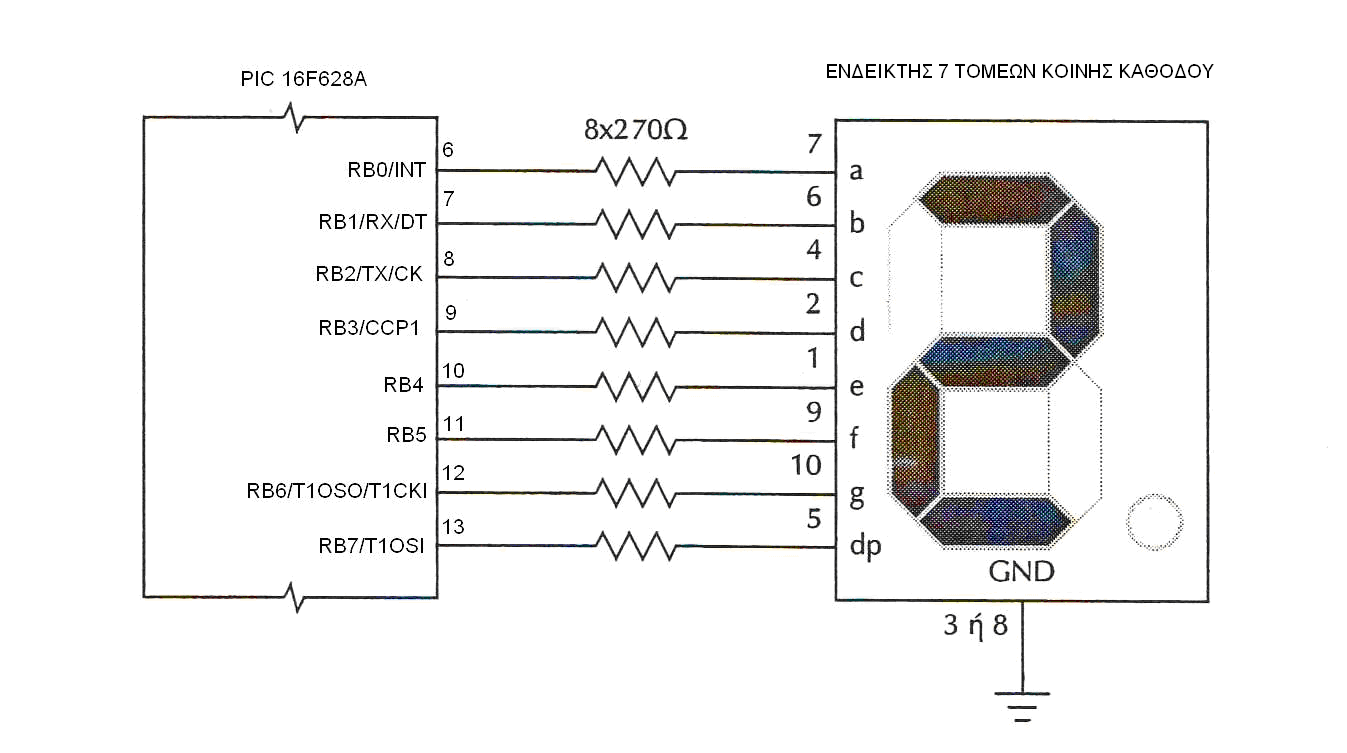
*Οι παραπάνω αλλαγές στον ακροδέκτη RD0 μπορούν να επαναληφθούν με τα ίδια αποτελέσματα.*



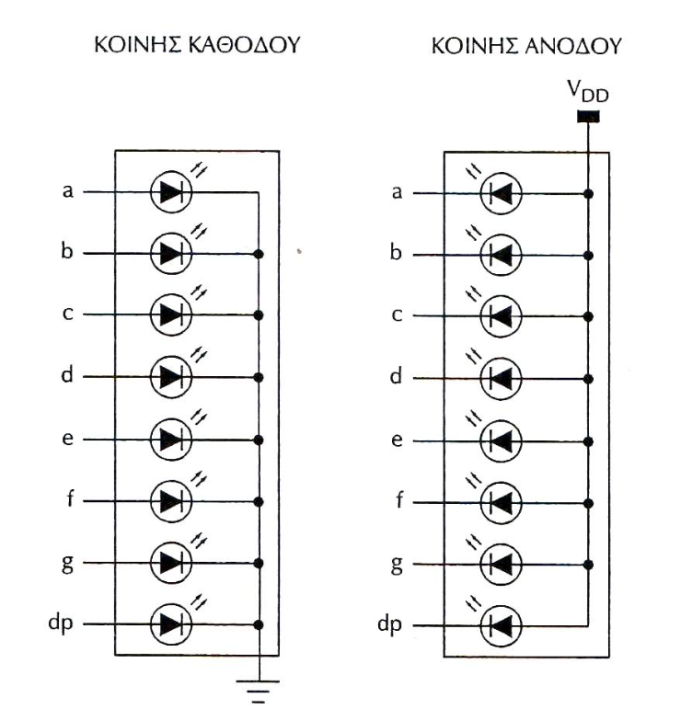
Κύκλωμα άσκησης 3Γ



*Συνδέσεις των ενδεικτών 7 τομέων*



*Συνδέσεις των ενδεικτών 7 τομέων*



*Εσωτερική δομή ενδειτκών 7 τομέων κοινής καθόδου και κοινής ανόδου*

Διαγράμματα ροής:

Αρχή

Είναι D0=0 και δεν ξεκίνησε το πρόγραμμα;

NAI

ΟΧΙ

Είναι D0=1 και ξεκίνησε το πρόγραμμα(start=1);

NAI

ΟΧΙ

stop=1 : Ο μετρητής χρόνου τρέχει.

Ο ενδείκτης εμφανίζει την τιμή χρόνου κατά την στιγμή μετάβασης του ακροδέκτη RD0 από 0 σε 1

stop=0: Ο μετρητής χρόνου τρέχει.

Ο ενδείκτης εμφανίζει την τρεχούμενη τιμή χρόνου

Ξεκίνα τον μετρητή(start=1)

Κύριο πρόγραμμα

Ρουτίνα διακοπής από τον Timer0

Αρχή ρουτίνας διακοπής από τον timer0

Ελάττωσε τον μετρητή διακοπών (counter) κατά 1

Είναι ο counter=0;

ΝΑΙ

ΟΧΙ

Ξεκίνησε ο μετρητής δεκάτων δευτερολέπτου(start=1) ;

ΝΑΙ

ΟΧΙ

Αύξησε τον μετρητή(δεκάτων του δευτερολέπτου) κατά 1

Απόδοση της τιμής 20 στον μετρητή διακοπών counter. Κάθε 20 διακοπές έχει περάσει 0,1 sec(20X5ms=100ms=0,1 s).

Αν ο μετρητής(δεκάτων δευτερολέπτου) γίνει μεγαλύτερος από 999 ξαναπαίρνει την τιμή 0

Εμφάνιση τρεχούμενης τιμής στον ενδείκτη(stop=0);

ΝΑΙ

ΟΧΙ

Υπολογισμός εκατοντάδων, δεκάδων και μονάδων του μετρητή δεκάτων του δευτερολέπτου

Μεταφορά των τιμών στις αντίστοιχες μεταβλητές εκατοντάδων δεκάδων ή μονάδων

Απόδοση αρχικής τιμής στον timer0 έτσι ώστε να γίνονται διακοπές κάθε 20 msec

Επιλογή ενός μόνο ενδείκτη 7 τομέων για εμφάνιση του αντίστοιχου ψηφίου του μετρητή δεκάτων δευτερολέπτου.

Σε κάθε διακοπή(δηλαδή ανά 5 ms) εμφανίζεται μόνο ένα ψηφίο του μετρητή.

Δηλαδή το κάθε ψηφίο του μετρητή ανάβει κάθε 3 διακοπές

Εμφάνιση ενός ψηφίου του ενδείκτη 7 τομέων (εκατοντάδων, δεκάδων ή μονάδων)

Τέλος ρουτίνας διακοπής από τον timer0

Αρχή ρουτίνας αρχικοποίησης

Πόρτα Β έξοδος

Πόρτα C έξοδος

Πόρτα D είσοδος

Πόρτα Β έξοδος

Πόρτα C έξοδος

Πόρτα D είσοδος

Πόρτα Β=0, Πόρτα C=0

Μετρητής διακοπών(counter)=20

Μετρητής δεκάτων δευτερολέπτου(tenthseconds)=0

Μετρητής αποθήκευσης δεκάτων δευτερολέπτου (tenthseconds\_stop)=0

Διακόπτης εκκίνησης(start)=1

Διακόπτης εμφάνισης στους ενδείκτες(stop)=0

Μεταβλητή επιλογής ενδείκτη επτά τομέων(des)=0

Ενεργοποίηση του γενικού διακόπτη διακοπών

Ενεργοποίηση διακοπών από τον timer0

Ρύθμιση prescaler=1

Αρχική τιμή timer για διακοπές κάθε 5ms ίση με5536

Τέλος ρουτίνας αρχικοποίησης

Ρουτίνα αρχικοποίησης

Πρόγραμμα:

#include <main.h>

#use standard\_io ( A ) // Standard είσοδοι και έξοδοι // Standard είσοδοι και έξοδοι

#use standard\_io ( B)

#use standard\_io ( C )

#byte PORTB =0xF81 //Ορισμός των θυρών με την θέση τους στην μνήμη.

#byte PORTC =0xF82

#byte PORTD =0xF83

Δηλώσεις μεταβλητών

int8 des=0; //Μεταβλητή που χρησιμοποιείται από την πόρτα C για να ενεργοποιεί

// έναν κάθε φορά από τους ενδείκτες 1, 2, 3 αποστέλλοντας με την σειρά λογικό 1

// στις βάσεις των τρανζίστορ που συνδέουν τις καθόδους των τριών ενδεικτών

// προς την γη.

int16 tenthseconds,tenthseconds\_stop=0; //Με την μεταβλητή tenthseconds μετράμε δέκατα

//του δευτερολέπτου.

// Την μεταβλητή tenthseconds\_stop την χρησιμοποιούμε για

// να αποθηκεύσουμε την τιμή του χρόνου όταν ο

//ακροδέκτης RD0 μεταβεί από 0 σε 1.

int8 counter; // μεταβλητή για να μετράμε 20 διακοπές που συμβαίνουν κάθε 5 ms

//κάθε 20 διακοπές έχει περάσει ένα δέκατο του δευτερολέπτου(=100 ms)

int1 start=0; // Η μεταβλητή start γίνεται 1 όταν ξεκινήσει ο μετρητής και παραμένει για πάντα 1

int1 stop=0; // Η μεταβλητή stop παίρνει την τιμή 1 όταν γίνει RD0=1

// Τότε σταματάει να μεταφέρεται η τρέχουσα τιμή του μετρητή στους ενδείκτες.

// Οι ενδείκτες για stop =1 εμφανίζουν την τιμή κατά την στιγμή

//μετάβασης του RD0 από 0 σε 1. Δηλαδή όταν stop=1 «παγώνουν» οι ενδείκτες.

int8 table[16] ={ 0b00111111, //Πίνακας με του κώδικες για εμφάνιση σε ενδείκτη 7 τομέων

0b00000110, //των ψηφίων 0, 1, 2, … 9, … F.

0b01011011,

0b01001111,

0b01100110,

0b01101101,

0b01111101,

0b00000111,

0b01111111,

0b01101111,

0b01110111,

0b01111100,

0b00111001,

0b01011110,

0b01111001,

0b01110001};

int8 dig[3] = {1,2,4}; // πίνακας με τις τιμές 0000 0001, 0000 0010, 0000 0100 για ενεργοποίηση

// από τους ακροδέκτες της πόρτας C με την σειρά ενός κάθε φορά από

// τους τρεις ενδείκτες κοινής καθόδου. Εφαρμόζονται 5 V στην βάση του

//αντίστοιχου τρανζίστορ και αυτό έρχεται στον κόρο συνδέοντας την κοινή

//κάθοδο του ενδείκτη προς την γη.

// Δήλωση συναρτήσεων

void timer0\_int(void);

void init (void);

// Κύρια συνάρτηση

void main()

{

init(); //ρουτίνα αρχικοποίησης με αρχικές τιμές

while (1){

if ((INPUT(PIN\_D0)==0) && (start==0)){ //Αυτή η συνθήκη ισχύει στο ξεκίνημα όταν RD0=0

// και start=0

start=1; // Στο ξεκίνημα γίνετα start=1 και δεν αλλάζει ποτέ.

}

else if(start == 1 && INPUT(PIN\_D0) == 1){ //Αυτό συμβαίνει όταν γίνει RD0=1

stop = 1; // Αν η μεταβλητή stop είναι 1

// οι ενδείκτες δεν εμφανίζουν

// την τρέχουσα τιμή χρόνου αλλά

// εμφανίζουν την τιμή χρόνου κατά την // προηγούμενη στιγμή μετάβασης

// του ακροδέκτη RD0 από 0 σε 1

}

else { //Αυτό συμβαίνει όταν RD0=0

stop = 0; //Οι ενδείκτες εμφανίζουν την

// τρέχουσα τιμή χρόνου.

}

} //κλείνει η while

} //κλείνει η main

// Ορισμός συναρτήσεων

#INT\_TIMER0 HIGH // Διακοπή με μεγάλη προτεραιότητα από τον Timer0

void timer0\_int(void){

int16 mon,dec,eka; // Μεταβλητές για αποθήκευση μονάδων, δεκάδων, εκατοντάδων

counter--; // Κάθε 20 \* 5 msec = 0,1 sec

// Διακοπές συμβαίνουν κάθε 5 ms. O counter παίρνει την αρχική τιμή 20

// Κάθε 20 διακοπές έχουν περάσει 20Χ5ms=100ms=0,1 s.

if (counter == 0){ // κάθε φορά που μηδενίζεται ο counter έχουν περάσει 0,1 s

if (start == 1){ //Αν τρέχει ο μετρητής

tenthseconds++; // αυξάνεται κατά 1 κάθε 0,1 sec(=20 διακοπές)

}

counter = 20; //αν γίνουν 20 διακοπές, δηλαδή counter=0

//ο μετρητής διακοπών ξαναπαίρνει την τιμή 20

if (tenthseconds > 999){ // Ο μετρητής μετά την τιμή 999

tenthseconds = 0; // ξαναπαίρνει την τιμή 0.

}

}

if (stop == 0){

tenthseconds\_stop = tenthseconds; // Όταν είναι stop=0(δηλαδή RD0=0)

//αποθηκεύεται συνεχώς η τρέχουσα

//τιμή του μετρητή στην

// μεταβλητή seconds\_stop.

// Η ένδειξη του μετρητή θα τρέχει

// Όταν γίνει stop=1(δηλαδή RD0=1) η μεταβλητή

//tenthseconds κρατάει σταθερά την τιμή

//που είχε κατά την στιγμή της μετάβασης

//του ακροδέκτη RD0 από 0 σε 1

// Επομένως όταν RD0=1 η ένδειξη θα παραμένει

// παγωμένη(σταθερή)

}

eka = (int8) (tenthseconds\_stop /100); //Υπολογισμός των εκατοντάδων του μετρητή

dec = (int8) ((tenthseconds\_stop - (100\*eka))/10); //Υπολογισμός των δεκάδων του

// μετρητή

mon = (int8) (tenthseconds\_stop - (100 \* eka) -(10 \* dec)); //Υπολογισμός των μονάδων

// του μετρητή

set\_timer0(5536); // Αρχική τιμή του timer0 για να έχουμε διακοπές κάθε 5 ms

des = ++des%3; //Μεταβλητή des παίρνει τις τιμές 0, 1, 2, 0, 1, 2 ,0, 1, 2,…

//%3 σημαίνει modulo 3(υπόλοιπο διαίρεσης με το 3)

PORTC = dig[des]; //Η πόρτα C παίρνει τιμές 0000 0001, 0000 0010, 0000 0100,…

// δηλαδή τις τιμές 1, 2, 4, 1, 2, 4 ,… από τον πίνακα dig

//Η πόρτα C επιλέγει σε ποιος ενδείκτης θα ανάψει.

// Εμφάνιση μονάδων, δεκάδων, εκατοντάδων στους ενδείκτες

if (des==0){ //PORTC=0000 0001(=dig[0]), επιλογή ενδείκτη μονάδων

PORTB = table[mon]; //Αν des=0 ενεργοποιείται ο ενδείκτης των μονάδων

//PORTC=0000 0001(=dig[0])

}

if (des==1){ //PORTC=0000 0010(=dig[1]), επιλογή ενδείκτη δεκάδων

PORTB = table[dec]; //Αν des=1 ενεργοποιείται ο ενδείκτης των δεκάδων

}

if (des==2){ //PORTC=0000 0010(=dig[2]), επιλογή ενδείκτη εκατοντάδων

PORTB = table[eka]; //Αν des=2 ενεργοποιείται ο ενδείκτης των εκατοντάδων

//PORTC=0000 0100(=dig[2])

}

}

// Συνάρτηση αρχικοποίησης

void init (void){

set\_tris\_b(0x00); // Καθορισμός της πόρτας Β ως έξοδος

set\_tris\_c(0x00); // Καθορισμός της πόρτας C ως έξοδος

set\_tris\_d(0x0f); // Καθορισμός της πόρτας C ως έξοδος

PORTB = 0; //Αρχική τιμή πόρτας Β

PORTC = 0; // Αρχική τιμή πόρτας C

counter = 20; // Αρχική τιμή του counter

tenthseconds = 0; //Αρχική τιμή μετρητή δεκάτων δευτερολέπτου

tenthseconds\_stop = 0; //Αρχική τιμή της μεταβλητής tenthseconds\_stop

start=0; //κατά την εκκίνηση start=0. Στην συνέχεια γίνεται 1 και παραμένει πάντα 1

stop=0; //αρχική τιμή μεταβλητής stop=0, δηλαδή ο μετρητής εμφανίζεται στους ενδείκτες.

des =0; //αρχική τιμή μεταβλητής που χρησιμοποιείται για επιλογή ενδείκτη από την θύρα C

SETUP\_TIMER\_0(T0\_INTERNAL | T0\_DIV\_1 ); //Ρύθμιση prescaler=1

set\_timer0(5536); // Αρχική τιμή του timer για διακοπές κάθε 5 ms

enable\_interrupts(INT\_TIMER0); // Ενεργοποίηση της διακοπής του timer0

enable\_interrupts(GLOBAL); // Ενεργοποίηση του γενικού // διακόπτη των διακοπών

} // κλείνει η αγκύλη της init()