

ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

**Output 2: Online Course for Microcontrollers:
syllabus, open educational resources**

Practice leaflet: Module_2-7 Ενδείκτης 7 τομέων

Lead Partner: International Hellenic University (IHU)

Δήλωση

Αυτό το αρχείο συντάχθηκε στο πλαίσιο του έργου ENGINE. Όπου έχουν χρησιμοποιηθεί άλλα δημοσιευμένα και αδημοσίευτα υλικά, αυτά έχουν αναγνωρισθεί.

Πνευματική ιδιοκτησία

© Copyright 2021 - 2023 the [ENGINE](#) Consortium

Warsaw University of Technology (Poland)

International Hellenic University (IHU) (Greece)

European Lab for Educational Technology- EDUMOTIVA (Greece)

University of Padova (Italy)

University of Applied Sciences in Tarnow (Poland)

Όλα τα δικαιώματα διατηρούνται.



Αυτό το έγγραφο έχει άδεια Creative Commons Attribution-NonCommercial- NoDerivatives 4.0 International License.

Αυτό το έργο έχει χρηματοδοτηθεί με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Αυτή η έκθεση αντικατοπτρίζει μόνο τις απόψεις του συγγραφέα και η Επιτροπή δεν μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν.

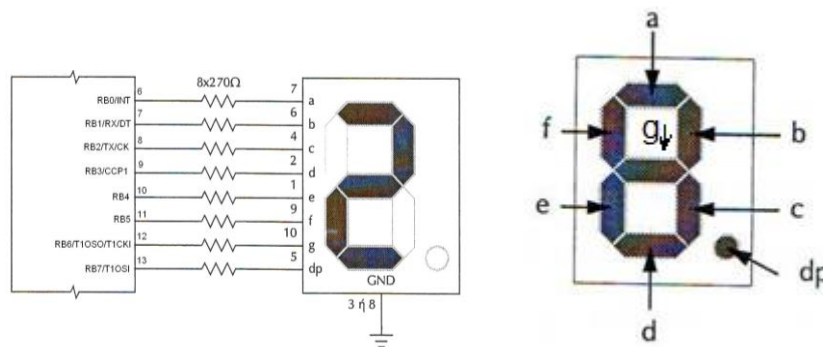
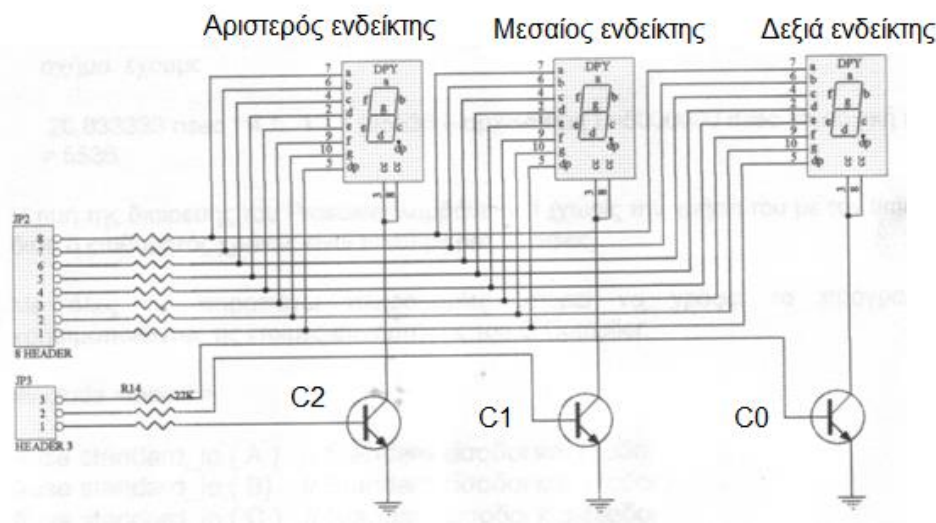
Πίνακας Περιεχομένων

Δραστηριότητες.....	4
1. Εμφάνιση τιμής σε ενδείκτες 7 τομέων.....	4
2. Μετατροπή και εμφάνιση δυαδικού αριθμού εισόδου, σε δεκαδικό αριθμό με χρήση ενδεικτών 7 τομέων	6
3. Δημιουργία ρολογιού πραγματικού χρόνου με ενδείκτες 7 τομέων.....	8

Δραστηριότητες

1. Εμφάνιση τιμής σε ενδείκτες 7 τομέων

Τρεις ενδείκτες 7 τομέων κοινής καθόδου συνδέονται στην πόρτα Β του μικροελεγκτή 18F4550 όπως φαίνεται στο σχήμα. Η βάση του τρανζίστορ που αντιστοιχεί στον δεξιά ενδείκτη συνδέεται στον ακροδέκτη C0, η βάση του τρανζίστορ που αντιστοιχεί στον μεσαίο ενδείκτη συνδέεται στον ακροδέκτη C1 και η βάση του τρανζίστορ που αντιστοιχεί αριστερά ενδείκτη συνδέεται στον ακροδέκτη C2.



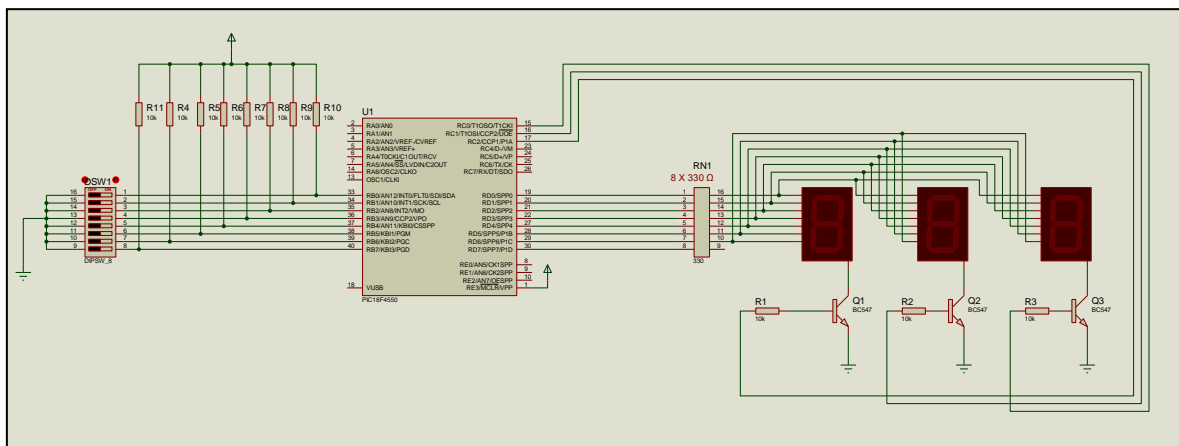
Σκοπός της δραστηριότητας είναι να εμφανιστεί ο αριθμός 12.3 στους ενδείκτες.

(35 λεπτά)	Βήμα 1. Υλοποίηση του κυκλώματος
	Βήμα 2. Ολοκλήρωση κώδικα
	Βήμα 3. Μεταφορά κώδικα στον μικροελεγκτή
	Βήμα 4. Έλεγχος λειτουργίας
	Βήμα 5. Τροποποιήσεις

	Στο Proteus η εμφάνιση κάθε ψηφίου πρέπει να τεθεί στα 100ms.
Βήμα 3 (1 λεπτά)	Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή
Βήμα 4 (1 λεπτά)	Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά
Βήμα 5 (15 λεπτά)	<ol style="list-style-type: none"> 1. Τροποποιήστε το πρόγραμμα της άσκησης ώστε στους τρεις ενδείκτες 7 τομέων να εμφανίζεται η λέξη HGE. 2. Αλλάξτε τον χρόνο εμφάνισης του κάθε ψηφίου από 5ms σε 200ms. Τι αποτέλεσμα έχει αυτή η αλλαγή στην εμφάνιση των ενδεικτών;

2. Μετατροπή και εμφάνιση δυαδικού αριθμού εισόδου, σε δεκαδικό αριθμό με χρήση ενδεικτών 7 τομέων

Να διαβάζεται συνεχώς η τιμή της πόρτας B που χρησιμοποιείται σαν είσοδος και να εμφανίζεται η τιμή της στο δεκαδικό αριθμητικό σύστημα στους 3 ενδείκτες 7 τομέων.



(25 λεπτά)	<p>Βήμα 1. Υλοποίηση κυκλώματος</p> <p>Βήμα 2. Συμπλήρωση κώδικα</p> <p>Βήμα 3. Μεταφορά κώδικα στον μικροελεγκτή</p> <p>Βήμα 4. Έλεγχος λειτουργίας</p>
------------	--

<p>Βήμα 1 (5 λεπτά)</p>	<p>Υλοποίηση του κυκλώματος. Διασύνδεση PORTD με ενδείκτες και RC0, RC1, RC2 στα 3 τρανζίστορ. Η PORTB συνδέεται σε 8 διακοπτάκια</p>
<p>Βήμα 2 (15 λεπτά)</p>	<p>Συμπληρώστε τον παρακάτω κώδικα</p> <pre> #include <main.h> #define PORTB =0xF81 #define PORTC =0xF82 #define PORTD =0xF83 void main(){ // Καθορισμός της πόρτας B ως εισόδου // Καθορισμός της πόρτας C ως εξόδου // Καθορισμός της πόρτας D ως εξόδου ----- ----- ----- int8 table[16] ={ //Πίνακας με του κώδικες για εμφάνιση σε ενδείκτη 7 τομέων 0b00111111, //κώδικας για εμφάνιση του 0 0b00000110, //κώδικας για εμφάνιση του 1 0b01011011, //κώδικας για εμφάνιση του 2 0b01001111, //κώδικας για εμφάνιση του 3 0b01100110, //κώδικας για εμφάνιση του 4 0b01101101, //κώδικας για εμφάνιση του 5 0b01111101, //κώδικας για εμφάνιση του 6 0b00000111, //κώδικας για εμφάνιση του 7 0b01111111, //κώδικας για εμφάνιση του 8 0b01101111, //κώδικας για εμφάνιση του 9 0b01110111, //κώδικας για εμφάνιση του A 0b01111100, //κώδικας για εμφάνιση του B 0b00111001, //κώδικας για εμφάνιση του C 0b01011110, //κώδικας για εμφάνιση του D 0b01111001, //κώδικας για εμφάνιση του E 0b01110001}; //κώδικας για εμφάνιση του F int isodos; // μεταβλητή για αποθήκευση της τιμής εισόδου int monades; // μεταβλητή για αποθήκευση των μονάδων της τιμής εισόδου int decades; // μεταβλητή για αποθήκευση των δεκάδων της τιμής εισόδου int ekatontades; // τιμή για αποθήκευση των εκατοντάδων της τιμής εισόδου while (TRUE){ // οι κώδικες του κάθε ψηφίου στέλνονται κάθε 5 ms // με ενεργοποίηση του αντίστοιχου ενδείκτη //διαβάζουμε την τιμή της PORTB isodos=PORTB; // υπολογισμός των εκατοντάδων της τιμής της πόρτας B ekatontades=isodos/100; // υπολογισμός των δεκάδων της τιμής της πόρτας B decades=(isodos-ekatontades*100)/10; //υπολογισμός των μονάδων της τιμής της πόρτας B monades =isodos-ekatontades*100-decades*10; </pre>

	<pre> //.....Εμφάνιση των εκατοντάδων PORTC=0b00000100; //ενεργοποίηση του αριστερά ενδείκτη PORTD=table[ekatontades]; //αποστολή του κώδικα για εμφάνιση των εκατοντάδων delay_ms(5); //αναμονή για 5 ms //.....Εμφάνιση των δεκάδων PORTC=0b00000010; // ενεργοποίηση του μεσαίου ενδείκτη PORTD=table[decades]; //αποστολή του κώδικα για εμφάνιση των δεκάδων delay_ms(5); //αναμονή για 5 ms //.....Εμφάνιση των μονάδων ----- //ενεργοποίηση του δεξιά ενδείκτη ----- //αποστολή του κώδικα για εμφάνιση των μονάδων ----- //αναμονή για 5 ms } } Στο Proteus η εμφάνιση κάθε ψηφίου πρέπει να τεθεί στα 100ms. </pre>
<p>Βήμα 2 (3 λεπτά)</p>	<p>Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή</p>
<p>Βήμα 3 (2 λεπτά)</p>	<p>Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά</p>

3. Δημιουργία ρολογιού πραγματικού χρόνου με ενδείκτες 7 τομέων

Το σύστημα θα λειτουργεί ως εξής:

- Αρχικά η ένδειξη θα είναι 12:00. Επειδή υπάρχουν 3 ενδείκτες η ένδειξη θα γίνεται σε δύο φάσεις. Για ένα δευτερόλεπτο θα απεικονίζεται η ώρα και για ένα δευτερόλεπτο θα απεικονίζονται τα λεπτά.
- Για να μπορεί να γίνεται ο διαχωρισμός της ένδειξης μεταξύ των δύο ενδείξεων θα πρέπει όταν η ένδειξη στους δύο δεξιότερους ενδείκτες είναι η ώρα στον αριστερότερο ενδείκτη να εμφανίζεται η ένδειξη Ω ενώ όταν η ένδειξη είναι τα λεπτά στον αριστερότερο ενδείκτη να υπάρχει η ένδειξη Π.

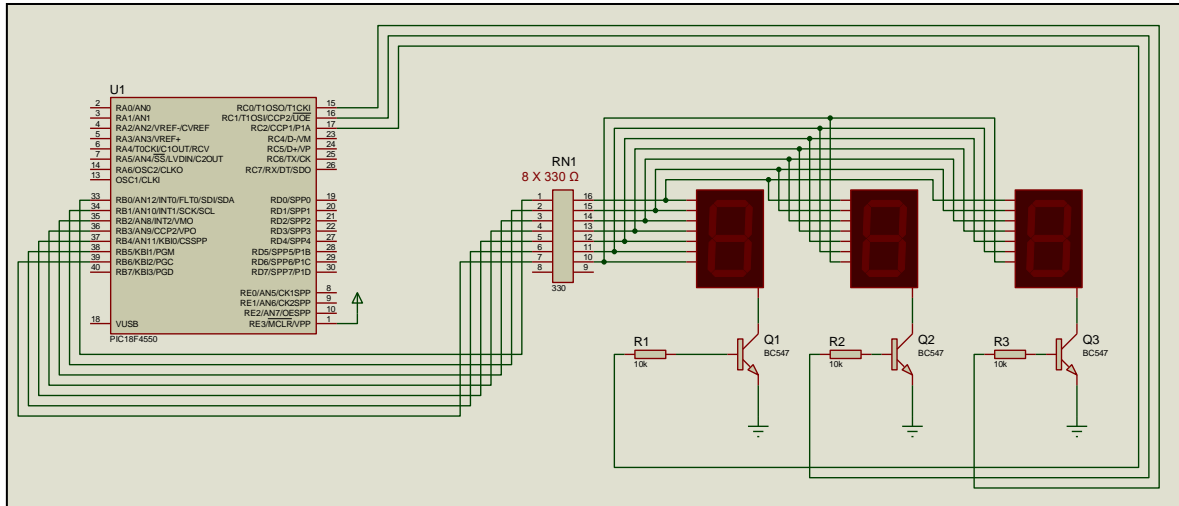
Να χρησιμοποιηθεί η μέθοδος της διακοπής από την υπερχειλίση του timer0 για την μέτρηση του χρόνου και οι πίνακες μετατροπής για την οδήγηση των ενδείκτων. **Συγκεκριμένα, ως βάση χρόνου θα χρησιμοποιηθούν τα 5ms, δηλαδή διακοπή από τον timer0 θα συμβαίνει κάθε 5ms. Όταν παρέλθουν 200 διακοπές, θα έχει συμπληρωθεί 1 δευτερόλεπτο.**

Για να υπολογίσουμε ποια τιμή πρέπει να πάρει ο timer0 πρέπει να λύσουμε την εξίσωση:

$$(65536 - y) * \frac{1}{\frac{F_{clock}}{4}} * Prescaler = 5ms$$

Έστω, Fclock=48MHz και Prescaler=1

Τότε προκύπτει πως $y \approx 5536$



<p>(25 λεπτά)</p>	<p>Βήμα 1. Υλοποίηση κυκλώματος</p> <p>Βήμα 2. Μεταφορά κώδικα στον μικροελεγκτή</p> <p>Βήμα 3. Έλεγχος λειτουργίας</p>
<p>Βήμα 1 (5 λεπτά)</p>	<p>Υλοποίηση του κυκλώματος. Διασύνδεση PORTB με ενδείκτες και RC0, RC1, RC2 στα 3 τρανζίστορ.</p>
<p>Βήμα 2 (15 λεπτά)</p>	<p>Μελετήστε τον παρακάτω κώδικα</p> <pre>#include <main.h> #byte PORTB =0xF81 #byte PORTC =0xF82 #byte PORTD =0xF83 // Ορισμοί μεταβλητών int8 des=0; //Μεταβλητή για επιλογή ενδείκτη 7 τομέων //από τον πίνακα int8 dig[3] μέσω της πόρτας C int8 seconds=0; //Μεταβλητή για την μέτρηση των δευτερολέπτων int8 minute=0; //Μεταβλητή για την μέτρηση των ωρών int8 hour=12;</pre>

```

//Μεταβλητή για την μέτρηση ωρών
int8 counter=200;
//Μεταβλητή για την μέτρηση διακοπών
int1 flag=0;
//Μεταβλητή για επιλογή εμφάνισης ωρών ή λεπτών

//Πίνακας για εμφάνιση των ψηφίων στους ενδείκτες
int8 table[16] = { 0b00111111, //0
                   0b00000110, //1
                   0b01011011, //2
                   0b01001111, //3
                   0b01100110, //4
                   0b01101101, //5
                   0b01111101, //6
                   0b00000111, //7
                   0b01111111, //8
                   0b01101111, //9
                   0b01101011, //Ω

                   //Ω Σύμβολο για την ώρα
                   0b00110111, //Π

                   //Π Σύμβολο για τα λεπτά
                   };

int8 dig[3] = {1,2,4};
//Πίνακας για ενεργοποίηση ενός μόνο ενδείκτη από την
//πόρτα C. Η πόρτα C εφαρμόζει 5V στην βάση ενός μόνον
//από τα τρία τρανζίστορ που συνδέουν την κοινή κάθοδο
//των ενδεικτών προς την γη.
//1 => 0000 0001, 2 => 0000 0010, 4 => 0000 0100

// Δήλωση συναρτήσεων
void timer0_int(void);
void init (void);
//Κύρια συνάρτηση
void main()
{
    init();
    while (1){
        // Το κύριο πρόγραμμα δεν κάνει τίποτα.
        // Περιμένει να συμβεί μια διακοπή
    }
}

// Ορισμός ρουτινών
// Ρουτίνα διακοπών
#INT_TIMER0 HIGH // Διακοπή με μεγάλη προτεραιότητα
void timer0_int(void){
    int16 mon,dec,eka;
//μεταβλητές για εμφάνιση ψηφίων
//στους ενδείκτες 7 τομέων
    set_timer0(5536);
// αρχική τιμή του μετρητή για να συμβαίνουν
//διακοπές κάθε 5 ms
    counter--;
//Ο μετρητής ελαττώνεται κατά 1 και μηδινίζεται
// κάθε 200 * 5 msec = 1 sec
    if (counter == 0){
        seconds++;
// Αυξάνεται ο μετρητής δευτερολέπτων κατά 1 κάθε
// 1 δευτερόλεπτο
        counter = 200;
// Αρχική τιμή του μετρητή διακοπών ώστε να μηδενίζεται
// μετά από 200 διακοπές δηλαδή μετά από 1 δευτερόλεπτο
//διότι 200 * 5 msec = 1 sec
        flag^=1;
// Ισοδυναμεί με το flag=flag^1.
// Σημαία που χρησιμοποιείται για να εμφανιστούν οι
// ώρες ή τα λεπτά

```

```

        if (seconds > 59){
            seconds = 0;
            // μετά την τιμή 59 ο μετρητής δευτερολέπτων παίρνει
            // την τιμή 0
            minute++;
            //Αν περάσουν 60 δευτερόλεπτα αυξάνει ο μετρητής
            // λεπτών κατά 1
            if (minute > 59){
                minute = 0;
                hour++;
            }
            //Αν περάσουν 60 λεπτά αυξάνει ο μετρητής ωρών κατά 1
        }

        if (hour >24){
            hour = 0;
            //Αν περάσουν 24 ώρες ο μετρητής ωρών γίνεται 0
        }
    }
    if (flag == 0){
        //αν το flag=0 τότε εμφανίζονται τα λεπτά
        dec = (int8)minute / 10;
        //Υπολογίζονται οι δεκάδες των λεπτών
        mon = minute - dec * 10;
        //Υπολογίζονται οι μονάδες των λεπτών
        eka = 11;
        // Στο 11ο στοιχείο το πίνακα int8 table[16] αντιστοιχεί
        // το σύμβολο για τα λεπτά «Π»
        // Το σύμβολο αυτό θα εμφανιστεί στον πρώτο από αριστερά
        // ενδείκτη(των εκατοντάδων)
    }
    if (flag == 1){
        //Αν το flag είναι 1 τότε εμφανίζονται οι ώρες
        dec = (int8)hour / 10;
        // υπολογισμός των δεκάδων των ωρών
        mon = hour - dec * 10;
        // υπολογισμός των μονάδων των ωρών
        eka = 10;
        //Στο 10ο στοιχείο το πίνακα int8 table[16]
        // αντιστοιχεί το σύμβολο για την ώρα, «Ω»
        // Το σύμβολο αυτό θα εμφανιστεί στον πρώτο από αριστερά
        // ενδείκτη(των εκατοντάδων)
    }

    des = ++des%3;
    /* Η μεταβλητή αυτή παίρνει διαδοχικά τις τιμές
    0, 1, 2, 0, 1, 2, 0, 1, ...
    ώστε να επιλέγονται από τον πίνακα dig[des] διαδοχικά
    οι τιμές ... 0000 0001, 0000 0010, 0000 0100 ...
    και να ενεργοποιούνται με την σειρά οι ενδείκτες */
    PORTC = dig[des];

    if (des==0){
        PORTB = table[mon];
        // εμφανίζονται οι μονάδες της ώρας ή των λεπτών
    }
    if (des==1){
        PORTB = table[dec];
        // εμφανίζονται οι μονάδες της ώρας ή των λεπτών
    }
    if (des==2){
        PORTB = table[eka];
        // εμφανίζεται ή ένδειξη ώρας «Ω» ή λεπτών «Π»
    }
}

// Ρουτίνα αρχικοποίησης
void init (void){
    set_tris_b(0x00);

```

	<pre> // Καθορισμός της πόρτας B ως εξόδου set_tris_c(0x00); // Καθορισμός της πόρτας C ως εξόδου PORTB = 0; // αρχική τιμή της πόρτας B PORTC = 0; // αρχική τιμή της πόρτας C counter = 200; // Αρχική τιμή του counter seconds = 0; // Αρχική τιμή του μετρητή δευτερολέπτων minute = 0; // Αρχική τιμή του μετρητή λεπτών hour = 12; // Αρχική τιμή του μετρητή ωρών des = 0; // Αρχική τιμή της μεταβλητής για επιλογή ενδείκτη flag = 0; // Αρχική τιμή της μεταβλητής για εμφάνιση ώρας ή λεπτών SETUP_TIMER_0(TO_INTERNAL TO_DIV_1); // Ρύθμιση prescaler=1 set_timer0(5536); // Αρχική τιμή του timer0 enable_interrupts(INT_TIMER0); // Ενεργοποίηση διακοπών από τον timer0 enable_interrupts(GLOBAL); // Ενεργοποίηση του γενικού διακόπτη των διακοπών } </pre>
<p>Βήμα 2 (3 λεπτά)</p>	<p>Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή</p>
<p>Βήμα 3 (2 λεπτά)</p>	<p>Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά</p>