

ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

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

Practice leaflet: Module_2-8 LCD16x2

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 μηνύματος σε LCD16x2.....	4
2. LCD16x2 και Timer0.....	5
3. Μέτρηση δευτερολέπτων και απεικόνιση σε LCD16x2	7

Δραστηριότητες

1. Εμφάνιση μηνύματος σε LCD16x2

Στόχος της δραστηριότητας είναι με την χρήση του αρχείου οδήγησης flex_lcd.h να εμφανίζεται στην 3η θέση της πρώτης γραμμής της LCD16x2 η λέξη ELECTRONICS και στην 4η θέση της δεύτερης γραμμής πέντε αστερίσκους.

(20 λεπτά)

Βήμα 1. Υλοποίηση του κυκλώματος

Βήμα 2. Ολοκλήρωση κώδικα

Βήμα 3. Μεταφορά κώδικα στον μικροελεγκτή

Βήμα 4. Έλεγχος λειτουργίας

Βήμα 1
(10 λεπτά)

Υλοποίηση του κυκλώματος. Διασύνδεση PORTB με LCD16x2.

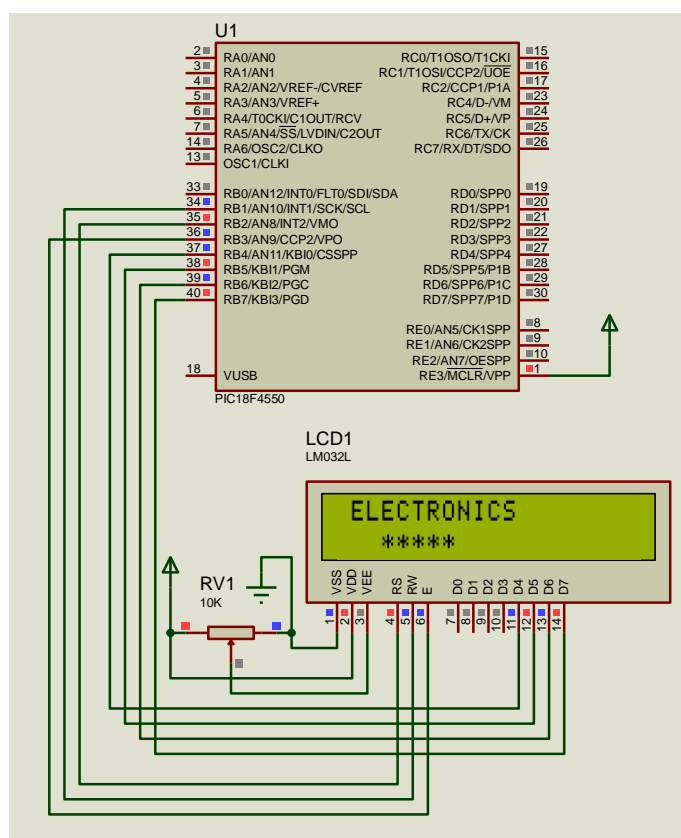


Figure 1. Διασύνδεση

<p>Βήμα 2 (5 λεπτά)</p>	<p>Μελετήστε τον παρακάτω κώδικα. Προσοχή! Ελέγξτε και διορθώστε αν χρειάζεται στον driver τα pins τα οποία χρησιμοποιούνται από τον μικροελεγκτή για την οθόνη.</p> <pre>#include <main.h> #include <flex_lcd.h> #byte PORTB=0xf81 void main() { lcd_init(); // αρχικοποίηση της οθόνης lcd_putc("\f"); // καθαρισμός της οθόνης lcd_gotoxy(3,1); // η οθόνη θα εμφανίσει στην 3η θέση της 1ης γραμμής // εμφανίζεται το μήνυμα lcd_putc(" ELECTRONICS"); lcd_gotoxy(4,2); // η οθόνη θα εμφανίσει στην 4η θέση της 2ης γραμμής lcd_putc("*****"); // εμφανίζεται το μήνυμα while(TRUE) {} }</pre>
<p>Βήμα 3 (3 λεπτά)</p>	<p>Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή</p>
<p>Βήμα 4 (2 λεπτά)</p>	<p>Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά</p>

2. LCD16x2 και Timer0

Σε αυτήν την δραστηριότητα θέλουμε να γραφεί πρόγραμμα το οποίο με την χρήση του αρχείου οδήγησης flex_lcd.h να εμφανίζεται στην 4η θέση της 1ης γραμμής της οθόνης LCD το μήνυμα MICRO LAB και στην 5η θέση της 2ης γραμμής να εμφανίζονται 6 αστερίσκοι οι οποίοι αναβοσβήνουν κάθε 200 ms. **Προσοχή το αναβοσβήσιμο να γίνεται με την βοήθεια ρουτίνας διακοπής από τον timer0!**

<p>(15 λεπτά)</p>	<p>*** το κύκλωμα είναι ίδιο με πριν***</p> <p>Βήμα 1. Συμπλήρωση κώδικα</p> <p>Βήμα 2. Μεταφορά κώδικα στον μικροελεγκτή</p> <p>Βήμα 3. Έλεγχος λειτουργίας</p>
-------------------	--

Βήμα 1
(11 λεπτά)

Συμπληρώστε τον παρακάτω κώδικα

```
#include <main.h>
#include <flex_lcd.h>

#define PORTB=0xf81

// Δήλωση μεταβλητών-----
//Μετρητής διακοπών (μετά από 4 διακοπές ανά 50 ms
θα έχουν περάσει 4X50=200 ms.
int counter=4;
// Μεταβλητή που δηλώνει το τι εμφανίζεται στην
οθόνη.
// state=0  εμφανίζονται οι αστερίσκοι
// state=1  δεν εμφανίζονται οι αστερίσκοι
int8 state;

// Δήλωση συναρτήσεων
void timer0_int(void);
void init (void);

void main() {
    //κλήση της ρουτίνας αρχικοποίησης
    init();
    // αρχικοποίηση της οθόνης
    lcd_init();
    // καθαρισμός της οθόνης
    lcd_putc("\f");
    // η οθόνη θα εμφανίσει στην 4η θέση της 1ης
γραμμής
    lcd_gotoxy(4,1);
    lcd_putc(" MICRO LAB"); // εμφανίζεται το
μήνυμα
    while(TRUE) {};
}

// Ρουτίνα διακοπής από τον timer0.
#define INT_TIMER0
void timer0_int(void){
    // αρχική τιμή του μετρητή για να συμβεί η
επόμενη διακοπή σε 50ms
    set_timer0(56161);
    counter--; // ελαττώνεται ο μετρητής διακοπών
    if (counter==0 && state==0 ) {//τύπωσε
αστερίσκους
        counter=4;
        state=1;
        lcd_gotoxy(5,2);
        lcd_putc("*****");
    }
    if (counter==0 && state==1) {//σβήσε
αστερίσκους
        -----
        -----
        -----
        -----
    }
}
```

	<pre>// Ρουτίνα αρχικοποίησης void init (void){ set_tris_b(0x00); PORTB = 0; state=0; counter=4; // Αρχική τιμή του counter //Prescaler=64 SETUP_TIMER_0(TO_INTERNAL TO_DIV_64); // Αρχική τιμή του μετρητή timer0 για διακοπές κάθε 50 ms set_timer0(56161); enable_interrupts(INT_TIMER0); enable_interrupts(GLOBAL); }</pre>
<p>Βήμα 2 (4 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. Μέτρηση δευτερολέπτων και απεικόνιση σε LCD16x2

Σε αυτήν την δραστηριότητα θέλουμε να γραφεί πρόγραμμα το οποίο θα μετράει δευτερόλεπτα ξεκινώντας από το 0. Στη θέση 5 της πρώτης γραμμής της οθόνης θα εμφανίζεται το μήνυμα SECONDS: και δίπλα η τιμή των δευτερολέπτων. **Προσοχή ο χρόνος θα υπολογίζεται με την βοήθεια ρουτίνας διακοπής από τον timer0!**

<p>(20 λεπτά)</p>	<p>*** το κύκλωμα είναι ίδιο με πριν***</p> <p>Βήμα 1. Ολοκλήρωση κώδικα</p> <p>Βήμα 2. Μεταφορά κώδικα στον μικροελεγκτή και έλεγχος λειτουργίας</p>
<p>Βήμα 1 (14 λεπτά)</p>	<p>Μελετήστε και συμπληρώστε τον παρακάτω κώδικα</p> <pre>#include <main.h> #include <flex_lcd.h> #byte PORTB=0xf81 //Δήλωση μεταβλητών //Μειρητής διακοπών (μετά από 20 διακοπές ανά 50 ms => 20X50=1000 ms=1 second. int counter=20; // μεταβλητή της οποίας η τιμή θα εμφανίζεται στην οθόνη</pre>

	<pre> int seconds=0; // Δήλωση συναρτήσεων void timer0_int(void); void init (void); void main() { init(); //κλήση της ρουτίνας αρχικοποίησης lcd_init(); //αρχικοποίηση της οθόνης lcd_putc("\f"); //καθαρισμός της οθόνης while(TRUE){;} } // Ρουτίνα διακοπής από τον timer0. #INT_TIMER0 void timer0_int(void){ // αρχική τιμή του μετρητή για να συμβεί η επόμενη διακοπή σε 50ms set_timer0(56161); // ελαττώνεται ο μετρητής διακοπών counter--; if (counter==0) { //όταν περάσει 1 δευτερόλεπτο counter=20; seconds++; lcd_gotoxy(5,1); printf(lcd_putc,"SECONDS=%d",seconds); } } // Ρουτίνα αρχικοποίησης void init (void){ set_tris_b(0x00); set_tris_d(0x00); // Αρχική τιμή του μετρητή δευτερολέπτων ίση με μηδέν seconds=0; // Αρχική τιμή του counter=20. Μετά από 20 διακοπές ο counter θα γίνει 0 counter=20; //Prescaler=1/64 SETUP_TIMER_0(T0_INTERNAL T0_DIV_64); // Αρχική τιμή του μετρητή timer0 για διακοπές κάθε 50 ms set_timer0(56161); // Ενεργοποίηση της διακοπής του timer0 ----- // Ενεργοποίηση του γενικού διακόπτη των διακοπών ----- } </pre>
<p>Βήμα 2 (6 λεπτά)</p>	<p>Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή. Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά</p>