## Άσκηση 7Γ. Ασκήσεις πάνω στην οθόνη LCD(εμφάνιση στην οθόνη με την βοήθεια της ρουτίνας διακοπής του timer0)

**Να γραφεί πρόγραμμα με το οποίο με την χρήση του προγράμματος οδήγησης flex\_lcd.h εμφανίζεται στην 4η θέση της 1ης γραμμής της οθόνης LCD το μήνυμα MICRO LAB και στην 5η θέση της 2ης γραμμής εμφανίζονται 6 αστερίσκοι οι οποίοι αναβοσβήνουν κάθε 200 ms.**

**Προσοχή! Το αναβοσβήσιμο να γίνεται με την βοήθεια ρουτίνας διακοπής από τον timer0.**



Κύκλωμα άσκησης 7Γ (χρήση της flex\_lcd.h). Αναβοσβήσιμο με τη βοήθεια της ρουτίνας διακοπής

**Προσοχή!!! Στον φάκελο όπου βρίσκεται το πρόγραμμα σας θα πρέπει να τοποθετείστε το αρχείο flex\_lcd.h**

#include <main.h>

#include <flex\_lcd.h> // Το h αρχείο του προγράμματος οδήγησης της οθόνης θα

// πρέπει να είναι στον ίδιο φάκελο όπου θα αποθηκεύσουμε

//το πρόγραμμα μας.

//θα πρέπει να γίνει έλεγχος και πιθανόν τροποποίηση

// στις δηλώσεις #define LCD\_DB4 PIN\_B4 κλπ που

// υπάρχουν στην flex\_lcd.c. Με αυτές τις

// δηλώσεις καθορίζεται σε ποιους ακροδέκτες του

// μικροελεγκτή συνδέεται ο κάθε ακροδέκτης της οθόνης.

// Ανοίξτε το flex\_lcd.h με το Notepad και κάνετε

//έλεγχο στις δηλώσεις σύνδεσης των ακροδεκτών. Αν δεν

//συμφωνούν με τον τρόπο που συνδέσατε τον μικροελεγκτή στην

//οθόνη θα πρέπει να κάνετε τροποποιήσεις.

#byte PORTB=0xf81 // στην θέση 0xf81 είναι ο καταχωρητής δεδομένων

//της πόρτας Β

// Δήλωση μεταβλητών

int counter=4; //Μετρητής διακοπών (μετά από 4 διακοπές ανά 50 ms θα έχουν

// περάσει 4Χ50=200 ms.

int8 state=0; // Μεταβλητή που δηλώνει το τι εμφανίζεται στην οθόνη

// state=1 σημαίνει εμφανίζονται οι αστερίσκοι

// state=0 σημαίνει δεν εμφανίζονται οι αστερίσκοι

// Δήλωση συναρτήσεων

void timer0\_int(void); //Οι συναρτήσεις που θα χρησιμοποιηθούν δηλώνονται πάνω από την main()

void init (void);

void main() {

init(); //κλήση της ρουτίνας αρχικοποίησης

lcd\_init(); // αρχικοποίηση της οθόνης

lcd\_putc("\f"); // καθαρισμός της οθόνης

lcd\_gotoxy(4,1); // η οθόνη θα εμφανίσει στην 4η θέση της 1ης γραμμής

lcd\_putc("MICRO LAB"); // εμφανίζεται το μήνυμα στην θέση που

// ορίσθηκε από την προηγούμενη εντολή

while(TRUE) //ατέρμων βρόχος(βρόχος που εκτελείται συνεχώς χωρίς καμιά εντολή).

{ }

} // κλείνει η αγκύλη του main

// Ρουτίνα διακοπής από τον timer0.

#INT\_TIMER0 // Directive. Οδηγία προς τον Compiler. Δηλώνει ότι η επόμενη ρουτίνα

// είναι ρουτίνα

// διακοπών από τον timer0

void timer0\_int(void){

set\_timer0(56161); // αρχική τιμή του μετρητή

// για να συμβεί η επόμενη

// διακοπή σε 50ms

counter--; // ελαττώνεται ο μετρητής διακοπών

if (counter==0 && state==0 ) { // Αυτό συμβαίνει 4 διακοπές(=4X50ms=200 ms) μετά

//το σβήσιμο των αστερίσκων

**Να συμπληρωθούν οι απαραίτητες εντολές, δηλαδή να δίνεται νέα τιμή**

**στον counter, να αλλάζει η κατάσταση state και να εμφανίζονται οι αστερίσκοι.**

}

if (counter==0 && state==1) { // Αυτό συμβαίνει 4 διακοπές(=4X50ms=200 ms) μετά

//την εμφάνιση των αστερίσκων

**Να συμπληρωθούν οι απαραίτητες εντολές, δηλαδή να δίνεται νέα τιμή**

**στον counter, να αλλάζει η κατάσταση state και να σβήνονται οι αστερίσκοι.**

}

} //αγκύλη κλεισίματος της ρουτίνας διακοπών

// Ρουτίνα αρχικοποίησης

void init (void){

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

set\_tris\_d(0x00); // Καθορισμός της πόρτας D ως εισόδου

PORTB = 0;

state=0;

counter=4; // Αρχική τιμή του counter που αντιστοιχεί σε RD0=0

SETUP\_TIMER\_0(T0\_INTERNAL | T0\_DIV\_64 ); //Prescaler=

set\_timer0(56161); // Αρχική τιμή του μετρητή timer0 για διακοπές

//κάθε 50 ms

enable\_interrupts(INT\_TIMER0); // Ενεργοποίηση της

//διακοπής του timer0

enable\_interrupts(GLOBAL); // Ενεργοποίηση του γενικού

// διακόπτη των διακοπών

}