## Άσκηση 8Β. Πραγματοποίηση ρολογιού στην οθόνη LCD με δυνατότητα ρύθμισης της ώρας.

**Να πραγματοποιηθεί ρολόι στην οθόνη LCD με την χρήση διακοπών από τον χρονιστή timer0. Θα χρησιμοποιηθεί ο οδηγός(driver) flex\_lcd.h για την οδήγηση της οθόνης. Με την παρακάτω εντολή του οδηγού flex\_lcd.h εμφανίζονται ακέραιοι αριθμοί στην οθόνη. Να υπάρχει δυνατότητα ρύθμισης της ώρας με την βοήθεια των διακοπτών που συνδέονται στην πόρτα D όπως φαίνεται στο σχήμα.**

**‘Όταν η πόρτα D έχει την τιμή 1111 1110 θα σταματάει το ρολόι και θα αυξάνονται τα λεπτά κάθε 1 δευτερόλεπτο ώστε να μπορεί να γίνει η ρύθμιση τους.**

**‘Όταν η πόρτα D έχει την τιμή 1111 1101 θα σταματάει το ρολόι και θα αυξάνονται οι ώρες κάθε 1 δευτερόλεπτο ώστε να μπορεί να γίνει η ρύθμιση τους.**

**‘Όταν η πόρτα D έχει την τιμή 1111 1011 θα σταματάει το ρολόι και θα μηδενίζονται τα δευτερόλεπτα.**



Κύκλωμα της άσκησης 8Β. Ρολόι με τη χρήση του εσωτερικού ρολογιού του μικροελεγκτή και δυνατότητα ρύθμισης της ώρας

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

**Πρόγραμμα:**

#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

#byte PORTD=0xf83 // στην θέση 0xf83 είναι ο καταχωρητής δεδομένων

//της πόρτας D

**// Δήλωση μεταβλητών**

int counter=20; //Μετρητής διακοπών (μετά από 20 διακοπές ανά 50 ms θα έχουν

// περάσει 20Χ50=1000 ms=1 second.

int seconds=0; // μεταβλητή για μέτρηση δευτερολέπτων

int minutes=31; // μεταβλητή για μέτρηση λεπτών, εδώ δίνεται μια αυθαίρετη αρχική τιμή

int hours=12; // μεταβλητή για μέτρηση ωρών, εδώ δίνεται μια αυθαίρετη αρχική τιμή

**// Δήλωση συναρτήσεων**

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

void init (void);

void show\_separators(void); //Ρουτίνα εμφάνισης διαχωριστικών : :

void show\_seconds(void); // Ρουτίνα εμφάνισης δευτερολέπτων

void show\_minutes(void); // Ρουτίνα εμφάνισης λεπτών

void show\_hours(void); // Ρουτίνα εμφάνισης ωρών

**// Κύριο πρόγραμμα**

void main() {

init(); //κλήση της ρουτίνας αρχικοποίησης

lcd\_init(); // αρχικοποίηση της οθόνης

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

show\_separators(); // εμφάνιση των διαχωριστικών

show\_minutes();show\_seconds();show\_hours(); //εμφάνιση της αρχικής τιμής ώρας, λεπτών,

// δευτερολέπτων.

**Να συμπληρωθούν οι εντολές μέσα στην while(TRUE).**

**Για βοήθεια υπάρχουν τα σχόλια, τα οποία δεν είναι**

**υποχρεωτικό να ακολουθείστε, αρκεί να πετύχετε το**

**επιθυμητό αποτέλεσμα.**

while(TRUE) //ατέρμων βρόχος

{

if (PORTD==0b11111110) { // Αν η πόρτα D έχει την τιμή 11111110

// Απενεργοποίηση των

//διακοπών από τον timer0

//Αύξησε τα λεπτά κατά 1

//Αν τα λεπτά είναι λιγότερα από 60 εμφάνισε τα λεπτά

//Διαφορετικά μηδένισε τα λεπτά, εμφάνισε τα λεπτά

//αναμονή για ένα δευτερόλεπτο

} //κλείνει η αγκύλη του if

else if (PORTD==0b11111101) { //Διαφορετικά αν(else if) η πόρτα D έχει την τιμή 11111101

// Απενεργοποίηση των

//διακοπών από τον timer0

//Αύξησε τις ώρες κατά 1

//Αν οι ώρες είναι λιγότερες από 24 εμφάνισε τις ώρες

//Διαφορετικά μηδένισε τις ώρες και εμφάνισε τες

//αναμονή για ένα δευτερόλεπτο

} //Κλείνει η αγκύκλη του else if

else if (PORTD==0b11111011) { //Διαφορετικά αν(else if) η πόρτα D έχει την τιμή 01111011

// Απενεργοποίηση των

//διακοπών από τον timer0

// μηδένισε τα δευτερόλεπτα και εμφάνισε τα.

} //Κλείνει η αγκύκλη του else if

else {

//Διαφορετικά(else) ενεργοποίηση των διακοπών από

//τον Timer0

} //κλείνει η αγκύλη του else

} // κλείνει η αγκύλη του while(TRUE)

} // κλείνει η αγκύλη του main

**// Ρουτίνα διακοπής από τον timer0.**

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

// είναι η ρουτίνα

// διακοπών από τον timer0

void timer0\_int(void){

set\_timer0(56121); // αρχική τιμή του μετρητή

// για να συμβεί η επόμενη

// διακοπή σε 50ms

counter--; // ελαττώνεται ο μετρητής διακοπών

if (counter==0) {

counter=20; //o counter ξαναπαίρνει την τιμή 20 για να μετρήσει άλλες 20 διακοπές

//Παρακάτω καταγράφεται το τι συμβαίναι αν περάσει ένα δευτερόλεπτο.

//Όταν ο μετρητής δευτερολέπτων(seconds) φτάσει το 60, τότε τα δευτερόλεπτα γίνονται 0 και

//ο μετρητής λεπτών αυξάνει κατά 1

//Όταν ο μετρητής λεπτών(minutes) φτάσει το 60, τότε τα λεπτά γίνονται 0 και ο μετρητής

//ωρών αυξάνει κατά 1

//Όταν ο μετρητής ωρών(hours) φτάσει το 24, τότε οι ώρες γίνονται μηδέν.

seconds++; //κάθε 20 διακοπές ο αριθμός των δευτερολέπτων αυξάνει κατά 1

if (seconds<60)

{show\_seconds();} // εμφάνιση των δευτερολέπτων

else {

seconds=0; // μηδενιδμός του μετρητή δευτερολέπτων όταν τα δευτερόλεπτα γίνουν 60

minutes++; // αύξηση των λεπτών όταν τα δευτερόλεπτα γίνουν 60

if(minutes<60)

{show\_minutes();show\_seconds();} // εμφάνιση λεπτών και δευτερολέπτων

else{ // Εδώ γράφουμε τι γίνεται όταν minutes=60

minutes=0; //μηδενισμός των λεπτών όταν τα λεπτά γίνουν 60

hours++; // αύξηση των ωρών όταν τα λεπτά γίνουν 60

if (hours<24)

{show\_minutes();show\_seconds();show\_hours();} // εμφάνιση των ωρών, λεπτών

// και δευτερολέπτων

else

hours=0; // μηδενισμός των ωρών όταν οι ώρες γίνουν 24

{show\_minutes();show\_seconds();show\_hours();} // εμφάνιση ωρών, λεπτών

// και δευτερολέπτων.

} // κλείνει η else όπου γράφουμε τι γίνεται όταν minutes=60

} // κλείνει η else όπου γράφουμε τι γίνεται όταν seconds=0

} // κλείνει η αγκύλη του if (counter==0)

} // κλείνει η ρουτίνα διακοπών από τον timer0

**// Ρουτίνα αρχικοποίησης**

void init (void){

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

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

seconds=0; // Αρχική τιμή του μετρητή δευτερολέπτων ίση με μηδέν

counter=20; // Αρχική τιμή του counter=20. Μετά από 20 διακοπές ο counter θα γίνει 0

SETUP\_TIMER\_0(T0\_INTERNAL | T0\_DIV\_64 ); //Prescaler=1/64

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

}

**//ρουνίνα για εμφάνιση των διαχωριστικών : :**

void show\_separators(void){

lcd\_gotoxy(3,1);

lcd\_putc(":");

lcd\_gotoxy(6,1);

lcd\_putc(":");

}

**//ρουτίνα εμφάνισης των δευτερολέπτων(seconds)**

void show\_seconds(void) {

lcd\_gotoxy(7,1); //εγγραφή στην 7η θέση της πρώτης γραμμής

printf(lcd\_putc,"%2d",seconds); // εμφάνιση των δευτερολέπτων σε 2 θέσεις στην οθόνη

}

**//ρουτίνα εμφάνισης των λεπτών(minutes)**

void show\_minutes(void) {

lcd\_gotoxy(4,1); //εγγραφή στην 4η θέση της πρώτης γραμμής

printf(lcd\_putc,"%2d",minutes); //εμφάνιση των λεπτών σε 2 θέσεις στην οθόνη

}

**//ρουτίνα εμφάνισης των ωρών(hours)**

void show\_hours(void) {

lcd\_gotoxy(1,1); //εγγραφή στην 1η θέση της πρώτης γραμμής

printf(lcd\_putc,"%2d",hours); //εμφάνιση των ωρών σε 2 θέσεις στην οθόνη

}