**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



Звіт до лабораторної роботи №3

з навчальної дисципліни: «Мікроконтролери ч.1»

**РОБОТА З ТАЙМЕРАМИ, LCD-ДИСПЛЕЄМ НА БАЗІ HD44780**

**ТА КЛАВІАТУРОЮ 4х4**

Варіант №11

**Виконав:**

ст. гр. ІР-22

Матвійчук А.І.

**Прийняв:**

Ст. викладач кафедри КСА

Лагун І.І.

Львів – 2021

**Порядок виконання роботи:**

1. Згідно варіанту завдання (таблиця 1) зібрати у пакеті симуляції

Proteus схему на основі МК ATmega2560 та написати програму

мовою C++ в Arduino IDE для реалізації вказаного завдання.

2. Розробити клієнтську програму для керування МК.

3. Протестувати роботу клієнтської програми зі схемою в Proteus.

4. Залити програму в Arduino-Kit.

5. Отримані результати представити викладачу.

**Завдання:**

1. Непарні номери варіантів використовують 8-ми розрядний

інтерфейс LCD для обміну даними, Парні номери варіантів

– 4-ри розрядний.

2. У завданні вказується таймер, за допомогою якого потрібно

реалізувати відлік інтервалів часу для зазначеного

пристрою.

3. Тривалість звукових сигналів вибирається студентом

самостійно, дотримуючись вказаних меж.

4. Натиск кожної кнопки супроводжується коротким

одинарним сигналом, успішне підтвердження введення

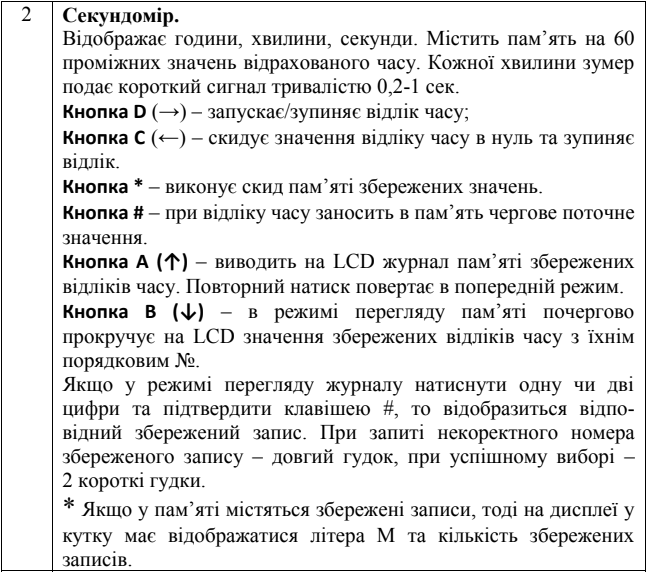
даних за допомогою клавіші # супроводжується коротким

подвійним сигналом, введення невірних даних за

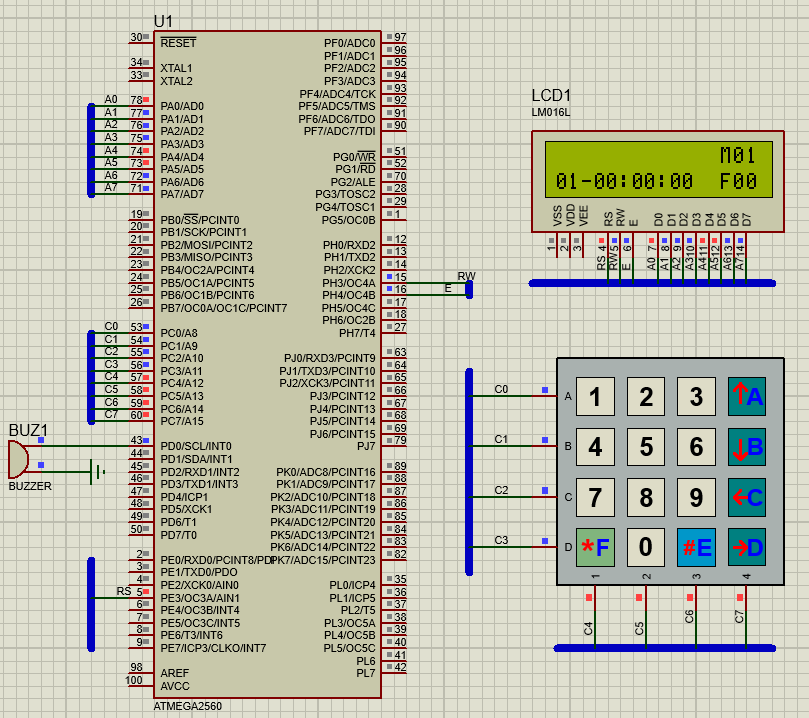
допомогою клавіші # та відміна усієї операції за допомогою

клавіші \* супроводжується довгим сигналом.

****

****

**Схема у Proteus:**



**Відео презентація лабораторної роботи:**

<https://drive.google.com/file/d/1bolL1pjYLKgv4G2Ot7YcucusihDqGg47/view>

**Код прошивки:**

**Lab3.cpp**

#include <LiquidCrystal.h>

#define DDR\_KEYPAD DDRC

#define PORT\_KEYPAD PORTC

#define PIN\_KEYPAD PINC

#include "keypad4x4.h"

const int buzzerPin = 21;

const int rs = 5, rw = 6, en = 7,d0 = 22, d1 = 23, d2 = 24,d3 = 25, d4 = 26, d5 = 27, d6 = 28, d7 = 29;

LiquidCrystal lcd(rs, rw, en,d0,d1,d2,d3,d4,d5,d6,d7);

const PROGMEM char sixty[60][3] = {

{"00"}, {"01"}, {"02"}, {"03"}, {"04"}, {"05"}, {"06"}, {"07"}, {"08"}, {"09"},

{"10"}, {"11"}, {"12"}, {"13"}, {"14"}, {"15"}, {"16"}, {"17"}, {"18"}, {"19"},

{"20"}, {"21"}, {"22"}, {"23"}, {"24"}, {"25"}, {"26"}, {"27"}, {"28"}, {"29"},

{"30"}, {"31"}, {"32"}, {"33"}, {"34"}, {"35"}, {"36"}, {"37"}, {"38"}, {"39"},

{"40"}, {"41"}, {"42"}, {"43"}, {"44"}, {"45"}, {"46"}, {"47"}, {"48"}, {"49"},

{"50"}, {"51"}, {"52"}, {"53"}, {"54"}, {"55"}, {"56"}, {"57"}, {"58"}, {"59"}

};

struct Time

{

unsigned char second, minute, hour;

};

Time T4 = {0, 0, 0};

Time arrTime[60];

byte countTime = 0;

boolean saveTime = 0;

boolean startCounting = 0;

boolean modeJournal = 0;

void LCD\_WriteStrPROGMEM(char \*str, int n)

{

for (int i = 0; i < n; i++)

lcd.print( (char)pgm\_read\_byte( &(str[i]) ) );

}

ISR(TIMER4\_COMPA\_vect)

{

if (startCounting){

if (++T4.second == 60)

{

T4.second = 0;

if (++T4.minute == 60)

{

T4.minute = 0;

if (++T4.hour == 24)

T4.hour = 0;

}

}

}

if(!modeJournal){

lcd.setCursor(3, 0);

LCD\_WriteStrPROGMEM(sixty[T4.hour], 2);

lcd.write(':');

LCD\_WriteStrPROGMEM(sixty[T4.minute], 2);

lcd.write(':');

LCD\_WriteStrPROGMEM(sixty[T4.second], 2);

}

}

void setup() {

noInterrupts(); // disable all interrupts

// Таймер#1: Скид при співпадінні OCR1A (1sec) + дільник=256

TCCR4A = 0x00;

TCCR4B = (1 << WGM12) | (1 << CS12) | (1 << CS10); //CTC mode & Prescaler @ 1024

TIMSK4 = (1 << OCIE4A); // дозвіл на переривання по співпадінню

OCR4A = 0x3D08;// compare value = 1 sec (16MHz AVR)

//KeyPad 4x4

initKeyPad();

//LCD 16x2

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

interrupts(); // Enable global interrupts

//Buzzer

pinMode(buzzerPin, OUTPUT);

digitalWrite(buzzerPin, LOW);

}

void loop() {

static byte scrollJournal = 1;

static byte numberJournal = 0;

static byte lastMinute = 0;

static byte newMinute = 0;

newMinute = T4.minute;

if (lastMinute!= newMinute){

lastMinute = newMinute;

buzzerSound(buzzerPin,200);

}

if ( isButtonPressed() ) {

lcd.setCursor(0, 1);

switch(readKeyFromPad4x4()){

if(!modeJournal){

case 'D':

startCounting = !startCounting;

saveTime = 0;

break;

case 'C':

startCounting = 0;

T4 = {0, 0, 0};

saveTime = 0;

break;

case 'F':

countTime = 0;

lcd.setCursor(13, 0);

lcd.write("DEL");

delay(1000);

lcd.setCursor(13, 0);

lcd.write(" ");

break;

}

case 'A':

lcd.clear();

if(countTime==0){

lcd.setCursor(4, 0);

lcd.write("NOTHING");

buzzerSound(buzzerPin,1000);

}

else{

modeJournal = !modeJournal;

if(modeJournal){

setJournal(1,countTime);

numberJournal = checkJournalNumber(0,0);

lcd.setCursor(13, 0);

lcd.write("M");

LCD\_WriteStrPROGMEM(sixty[countTime], 2);

}

else {

scrollJournal = 0;

lcd.setCursor(13, 0);

lcd.write("M");

LCD\_WriteStrPROGMEM(sixty[countTime], 2);

}

}

break;

case 'B':

if (scrollJournal<countTime){

scrollJournal++;

setJournal(scrollJournal,countTime);

}

break;

case '0': numberJournal = checkJournalNumber(numberJournal,0); break;

case '1': numberJournal = checkJournalNumber(numberJournal,1); break;

case '2': numberJournal = checkJournalNumber(numberJournal,2); break;

case '3': numberJournal = checkJournalNumber(numberJournal,3); break;

case '4': numberJournal = checkJournalNumber(numberJournal,4); break;

case '5': numberJournal = checkJournalNumber(numberJournal,5); break;

case '6': numberJournal = checkJournalNumber(numberJournal,6); break;

case '7': numberJournal = checkJournalNumber(numberJournal,7); break;

case '8': numberJournal = checkJournalNumber(numberJournal,8); break;

case '9': numberJournal = checkJournalNumber(numberJournal,9); break;

case 'E':

if(!modeJournal){

arrTime[countTime] = T4;

if (countTime<=60)countTime++;

lcd.setCursor(13, 0);

lcd.write("M");

LCD\_WriteStrPROGMEM(sixty[countTime], 2);

}

else{

if (numberJournal<=countTime && numberJournal!=0){

scrollJournal= numberJournal;

setJournal(numberJournal,countTime);

buzzerSound(buzzerPin,50);

delay(100);

buzzerSound(buzzerPin,50);

}

else{

buzzerSound(buzzerPin,500);

numberJournal = checkJournalNumber(numberJournal,0);

numberJournal = checkJournalNumber(numberJournal,0);

}

}

break;

}

}

}

byte checkJournalNumber(byte number,byte add){

byte newNumber = 0;

if (number>0) newNumber = number\*10 + add;

else if (number == 0) newNumber = add;

if (newNumber>59){

newNumber = 0;

}

lcd.setCursor(13, 1);

lcd.write("F");

LCD\_WriteStrPROGMEM(sixty[newNumber], 2);

return newNumber;

}

void setJournal(byte number, byte border){

if (number == 1){

setRowJournal(number,1);

lcd.setCursor(0, 0);

lcd.write(" ");

}

else if (number<=border){

setRowJournal(number-1,0);

setRowJournal(number,1);

}

}

void buzzerSound(byte pin,int timeDelay){

TIMSK4 &= ~(1 << OCIE1A);

digitalWrite(pin,HIGH);

delay(timeDelay);

digitalWrite(pin,LOW);

TIMSK4 = (1 << OCIE4A);

}

void setRowJournal(byte number, byte row){

lcd.setCursor(0, row);

LCD\_WriteStrPROGMEM(sixty[number], 2);

lcd.write('-');

LCD\_WriteStrPROGMEM(sixty[arrTime[number-1].hour], 2);

lcd.write(':');

LCD\_WriteStrPROGMEM(sixty[arrTime[number-1].minute], 2);

lcd.write(':');

LCD\_WriteStrPROGMEM(sixty[arrTime[number-1].second], 2);

}

**Keypad4x4.h**

//#define DDR\_KEYPAD DDRF

//#define PORT\_KEYPAD PORTF

//#define PIN\_KEYPAD PINF

const int Key\_1 = B11101110;

const int Key\_2 = B11011110;

const int Key\_3 = B10111110;

const int Key\_A = B01111110;

const int Key\_4 = B11101101;

const int Key\_5 = B11011101;

const int Key\_6 = B10111101;

const int Key\_B = B01111101;

const int Key\_7 = B11101011;

const int Key\_8 = B11011011;

const int Key\_9 = B10111011;

const int Key\_C = B01111011;

const int Key\_F = B11100111;

const int Key\_0 = B11010111;

const int Key\_E = B10110111;

const int Key\_D = B01110111;

unsigned char freePinFromKeyPad = 1;

unsigned char keyFromKeyPad = 1;

char pressedButton = 0;

bool logicPressed = false;

void initKeyPad() {

DDR\_KEYPAD = 0x0F;

PORT\_KEYPAD = 0xF0;

}

char readKeyFromPad4x4() {

logicPressed = false;

return pressedButton;

}

bool isButtonPressed() {

if (freePinFromKeyPad == 1) //перевірка чи була натиснута кнопка

{ //якщо =1, тоді ще не натискалася

if (PIN\_KEYPAD != 0xF0)

{

delay(50);

freePinFromKeyPad = 0;

keyFromKeyPad = 1;

// Визначення натиснутої клавіші

// почергова подача 0V на рядки клавіатури A,B,C,D

PORT\_KEYPAD = B11111110; // A-рядок

asm("nop");

if (PORT\_KEYPAD == PIN\_KEYPAD)

{

PORT\_KEYPAD = B11111101; // B-рядок

asm("nop");

if (PORT\_KEYPAD == PIN\_KEYPAD)

{

PORT\_KEYPAD = B11111011; // C-рядок

asm("nop");

if (PORT\_KEYPAD == PIN\_KEYPAD)

{

PORT\_KEYPAD = B11110111; // D-рядок

asm("nop");

if (PORT\_KEYPAD == PIN\_KEYPAD)

keyFromKeyPad = 0; // жодна клавіша не натиснута

}

}

}

if (keyFromKeyPad == 1) //визначення натиснутої клавіші

{

if (PIN\_KEYPAD == Key\_1) pressedButton = '1';

else if (PIN\_KEYPAD == Key\_2) pressedButton = '2';

else if (PIN\_KEYPAD == Key\_3) pressedButton = '3';

else if (PIN\_KEYPAD == Key\_4) pressedButton = '4';

else if (PIN\_KEYPAD == Key\_5) pressedButton = '5';

else if (PIN\_KEYPAD == Key\_6) pressedButton = '6';

else if (PIN\_KEYPAD == Key\_7) pressedButton = '7';

else if (PIN\_KEYPAD == Key\_8) pressedButton = '8';

else if (PIN\_KEYPAD == Key\_9) pressedButton = '9';

else if (PIN\_KEYPAD == Key\_0) pressedButton = '0';

else if (PIN\_KEYPAD == Key\_A) pressedButton = 'A';

else if (PIN\_KEYPAD == Key\_B) pressedButton = 'B';

else if (PIN\_KEYPAD == Key\_C) pressedButton = 'C';

else if (PIN\_KEYPAD == Key\_D) pressedButton = 'D';

else if (PIN\_KEYPAD == Key\_E) pressedButton = 'E';

else if (PIN\_KEYPAD == Key\_F) pressedButton = 'F';

logicPressed = true;

}

PORT\_KEYPAD = 0xF0; //відновлюємо порт

}

}

else if (PIN\_KEYPAD == 0xF0) //перевіряємо чи кнопка відпущена

{

delay(200);

freePinFromKeyPad = 1;

}

return logicPressed;

}

**Висновок:**

В даній лабораторній роботі я навчився використовувати таймери, використовувати графічний інтерфейс на LCD дисплеї і використовувати клавіатуру для зв’язку з інтерфейсом.